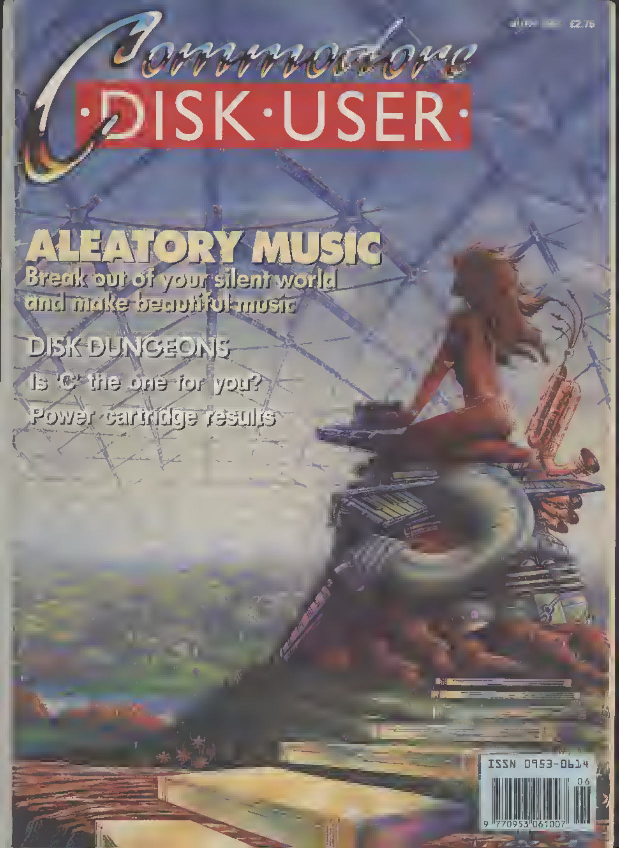
Commodore DISK·USER

[illegible] £2.75



ALEATORY MUSIC

Break out of your silent world and make beautiful music

DISK DUNGEONS

Is 'C' the one for you?

Power cartridge results

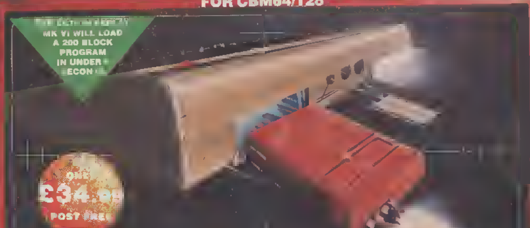
ISSN 0953-0614

06

THE ULTIMATE UTILITY CARTRIDGE COMES OF AGE!

NOW **ACTION REPLAY Mk VI** IS HERE

FOR CBM64/128



[illegible] RIGHT ACT
Datel Electronics neither condones or authorises the use of it's products for the reproduction of copyright material.
The back up facilities of this product are designed to reproduce only software such as public domain material, the users own programs or software where permission to make a back up has been clearly given.
It is illegal to make copies, even for your own use, of copyright material without the expressed permission of the copyright owner, or their licensee.

THE MOST POWERFUL FRIENDLY AND FEATURE PACKED UTILITY CARTRIDGE EVER CONCEIVED!

TURBO LOADER	Load [illegible] under six seconds - world's fastest disk serial loader. [illegible] loading speeds. Works with 1541/1571/Oceanic/[illegible]
INFINITE LIVES GENERATOR	[illegible] Very easy to use, [illegible] many [illegible]. No user knowledge [illegible]
PROF MACHINE CDE MNTR	Full 64K Freezer Monitor - examine ALL memory, including [illegible] I/O area and registers in [illegible] state. Ideal for de-bugging [illegible]
SPRITE CONTROL	Freeze the action and [illegible] [illegible]
FREEZER FACILITY	Now you can make your old slow loading programs load faster. Simply [illegible] the action and save to tape or disk to reload at [illegible] speed - no more waiting for programs to load.
DISK COPY	Easy to use disk/file copier. Much faster than conventional methods. Ideal for backing up [illegible]
TAPE TURBO	[illegible]
FAST FORMAT	[illegible]
PRINTER DUMP	[illegible]
CENTRONICS INTERFACE	For [illegible] Star, Epson etc. [illegible] (Cable [illegible] £12.99)
SCREEN EDITOR	Now you can edit the [illegible] "Great fun!"
EXTENDED TOOLKIT	Many single stroke commands for Load, Save, [illegible] range of [illegible] Auto Number, Delete, [illegible] Unnew, etc.

[illegible] REVIEWERS SAY
"I'm stunned amazed and totally impressed. This is really the best value for money cartridge The Cartridge King!"
[illegible]

[illegible]

HOW TO ORDER...

ALL ORDERS DESPATCHED WITHIN 48 HRS

PHONE	POST	FAX 0782 744292
24hr Credit Card Line	Send cheques/POs made payable to "Datel Electronics"	UK ORDERS POST FREE EUROPE ADD £1 OVERSEAS ADD £3

PRICES AND SPECIFICATIONS CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

DATEL ELECTRONICS

DATEL ELECTRONICS LTD., FENTON INDUSTRIAL ESTATE GOVAN ROAD, FENTON, STOKE-ON-TRENT, ENGLAND.

TECHNICAL PHONE LINE 0782 744324

GRAPHICS SUPPORT UTILITIES DISK

SLIDE SHOW View your favourite screens in a slide show type display

BLOW UP Unique utility allows you to take any part of a picture & 'blow it up' to full screen

SPRITE EDITOR A complete sprite editor helps you to create or edit sprites.

MESSAGE MAKER Any screen captured with Action Replay or created with a graphics package can be turned into a scrolling screen message with music

ONLY £9.99

AKVI

CONTENTS

Commodore Disk User
Volume 3 Number 8
June 1990

Editor: PAUL EVES
Group Editor: STUART COOKE
Production Editor: HILARY CURTIS
Photography: MANNY CEFAI
Adventure Correspondent: GORDON HAMLETT
Advertisement Manager: PAUL KAVANAGH
Display Sales Exec: MARIA WADE
Classified Sales Exec: TONY FLANAGAN
Designer: MARK NEWTON
Origination: EBONY TYPESETTING
Distribution: S M DISTRIBUTION
Printed By: CHASE WEB LTD ST IVES PLC

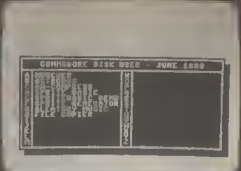
SUBSCRIPTION RATES

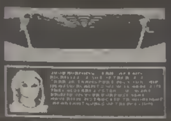
Here are the rates for subscriptions to CDU with effect from November 1989

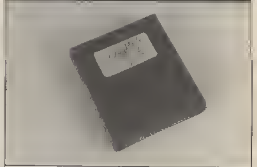
UK	**£33.00**
Europe	**£39.00**
Middle East	**£39.30**
Far East	**£41.60**
Rest of the World £39.70 or	**$69.00**

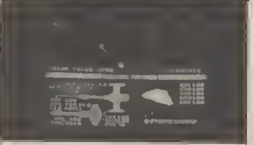
Airmail rates on request

Commodore Disk User is a monthly magazine published on the 3rd Friday of every month **Argus Specialist Publications, Argus House, Boundary Way, Hemel Hempstead, HP2 7ST. Telephone: (0442) 66551 Fax (0442) 66998**


COMMODORE DISK USER - JUNE 1990
MUNCHER
[illegible]
SPRITE BASIC
SPRITE GENERATOR
[illegible] MUSIC
FILE COPIER







A preview of what is in this month's issue

Opinions expressed in reviews are the opinions of the reviewers and not necessarily those of the magazine. While every effort is made to thoroughly check programs published we cannot be held responsible for any errors that do occur.

Editor's Comment

I am pleased to say that in general CDU is received very well by the majority of it's readers. I get letters by the dozen congratulating me on the overall appearance and content of the magazine and disk. It would seem that I have managed to achieve the aims that I set myself when I took over as Editor last September. That is to produce a magazine primarily for the serious computer user which is not only informative and full of technical help, but also contains sufficient programs for every walk of life, enabling enough interest to be generated to keep you buying your favourite magazine. I hope to be able to continue this trend. It is not all sugar and cream though, obviously I cannot hope to please 100 percent of you 100 percent of the time, (as Mr J M. Gatt will testify), so if you have a complaint or just want to gripe then let me hear from you. (Not too many now ..!!!)

After APRIL's edition, which most of you will be aware was devoted mainly to the new computer users amongst us, this months issue unfortunately comprises of nearly all written word and not a lot of program. I apologise for this here and now. The reason is because as some of you are aware, I have been fairly ill recently and am waiting to enter hospital for the third time this year. Hopefully, by the time this issue comes out I will be back to my normal self and thus so will the magazine. For all of you that have wished me well, I thank you, I hope the rest of you will bear with me for this month. I will have CDU back to it's usual standard commencing with the JULY issue.

Disk Instructions

We do our best to make sure that CDU will be compatible with all versions of the C64 and C128 computers. One point we must make clear is that the use of 'Fast Loaders', 'Cartridges' or alternative operating systems (Dolphin DOS) may not guarantee that your disk will function properly. If you use one or more of the above and you have difficulties, then I suggest you disable them and use the computer under normal, standard conditions. Getting the programs up and running should not present you with any difficulties, simply put your disk in the drive and enter the command.

LOAD"MENU",8,1

Once the disk menu has loaded you will be able to start any of the programs simply by pressing the letter that is to the left of the desired program. It is possible for some programs to alter the computer's memory so that you will not be able to LOAD programs from the menu correctly until you reset the machine. We therefore suggest that you turn your computer off and then on before loading each program.

How to copy CDU files

You are welcome to make as many of your own copies of CDU programs as you want, as long as you do not pass them on to other people, or worse, sell them for profit. For people who want to make legitimate copies, we have provided a simple machine code file copier. To use it, simply select the item FILE COPIER from the main menu. Instructions are presented on screen.

Disk Failure

If for any reason the disk with your copy of CDU will not work on your system then please carefully re-read the operating instructions in the magazine. If you still experience problems then:

1) If you are a subscriber, return it to:

Select Subscriptions Ltd
5, River Park Estate
Berkhamsted
HERTS HP4 1HL
Tele 0442-876661

2) If you bought it from a newsagents, then return it to:

CDU Replacements

Protoscan Europe PLC
Burrell Road
St Ives
Cambs
P17 4LE
Tele 0480-495520

(Within eight weeks of publication date disks are replaced free).

After eight weeks a replacement disk can be supplied from Protoscan for a service charge of £1.00. Return the faulty disk with a cheque or postal order made out of Protoscan and clearly state the issue of CDU that you require. No documentation will be provided.

Please use appropriate packaging, cardboard stiffener at least, when returning disk. Do not send back your magazine-only the disk please.

NOTE: Do not send your disks back to the above if its a program that does not appear to work. Only if the DISK is faulty. Program faults should be sent to the editorial office marked **FAO bug-finders**

Thank you.

BACK ISSUES BACK ISSUES
BACK ISSUES BACK ISSUES

Back issues of **CDU** are available at £3 25 per issue, which includes postage and packing via:

Select Subscriptions Ltd
5, River Park Estate
Berkhamsted
Herts
HP4 1HL
Tele: 0442-876661

Back Issues

VOL 2 No.5 JUL/AUG 89

FONT FACTORY – Create your own chars
HI-RES DEMO KIT – Add music to your favourite picture
ANIMATOR – Get those sprites moving
BORDER MESSAGE SCROLL – Say what you want along the bottom of the screen.
TYPIT 128 – Create professional text layouts on your C128.
SCREEN COPIES UTILITY – Download your favourite screens.
VIDI-BASIC – Graphic based extension to Basic.
64 NEWS DESK – Become a C64 reporter.

VOL 2 No.6 SEP/OCT 89

MICKMON – An extensive M/C monitor
SCRAPBOOK – Collectors and hobbyists database
CELLRATOR – Enter the caves if you dare
RAINBOW CHASER – Rainbows means points in the unusual game
HIDDEN GRAPHICS – Utilise those graphic secrets
FORTRESS – Save the world Yet again!!
DISK HUNTER – Keep tabs on your disk library.
SUPERFILE – One more for the record keepers.

VOL 3 No.1 NOVEMBER 89

BASIC EXTENSION – Windows and Icons the easy way
B-RAID – Vertical scrolling shoot 'em up.
DISKONOMISER – Prudent disk block saving
HELP – Design your own information help screens.
ORSITAL – An arcade style game with a difference
PROGRAM COMPARE – Modifying Basic progs has never been easier.
FASTER ROUTINES – A few colourful demos.
SPRITE EDITOR 1 – A no nonsense basic sprite editor
WABBIT – Help the rabbit collect his carrots.

VOL 3 No.3 JANUARY 90

4 IN A ROW – Connect a row of counters
FROGS IN SPACE – Leap to safety across the space lanes.
BLACKJACK – Don't lose your shirt.
LORD OF DARKNESS – Defeat the evil lord true adventure style
MARGO – Fly around and collect the jewels
JETRACE 2000 – Have you got what it takes to be best
ULTIMATE FONT EDITOR – Create your own screens and layouts.
SELECTIVE COLOUR RESTORE – Design your own start up colours.
6510+ UNASSEMBLER – Transform M/C into Source, with labels.
TRIVIA CHALLENGE – The first of 3 files for this superb game.
GEOS FONTS – The first 4 of 12 fonts for Geos users.

VOL 3 No.4 FEBRUARY 90

COLOUR PICTURE PRINT – Download your favourite colour screens.
BASE-ED 2 – An update to our popular database system.
1ST MILLION – Play the market in this strategy game.
FM-DOS – Enhance your drives operating system
GEOS FONTS – A further 4 fonts for Geos users
HASHING IT – Relative filing made easy
MULTI-SPRITE – Make full use of up to 24 sprites
DIRECTORIES EXPLAINED – Find your way through the directory jungle
TRIVIA CHALLENGE – The second part of this popular game.

VOL 3 No.5 MARCH 90

PLAGUE – Become your planets Guardian and Defender
SURROUND – Reversi on the C64
GEOS FONTS – The last of 12 new fonts.
SCREEN SLIDE – Create your own slideshows.
JOYSTICK TESTER – Put your stick(s) through the mill.
COLOUR MATCHER – Mastermind for the younger ones
SCREEN MANIPULATOR – Full use of the screen now obtainable.
TRIVIA CHALLENGE – The last of the 3 files for the game
VIDEO RECORDER PLANNER – Keep tab on your recordings.

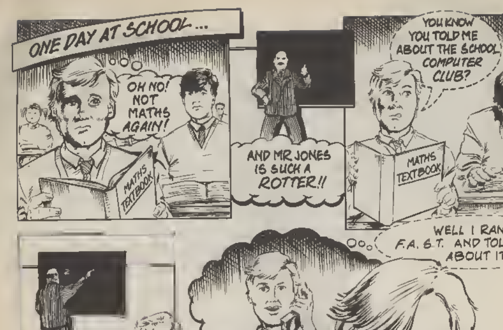
VOL 3 No. 6 APRIL 90

BAR PROMPTS – M/C alternative input routine
HI-LITE BARS – as Bar Prompts but in Basic.
TEXAS DEMO – Examples of using Basic for demos
CHARS TO SPRITES – Convert UDG's to sprites
FONT FACTORY – Compliments Chars to Sprites.
3D-TEXT MACHINE – Impressive 3D text screens made easy
SCREEN ENHANCER – Makes full use of the screen easy to do.
SPREADSHEET 64 – An excellent, easy to use spreadsheet.
MINI-AID – 3 short utilities to aid the Basic programmer.
C128 COLLECTION – 3 useful C128 programs

VOL 3 No.7 MAY 90

NUDGE – FLD explained and made easy.
WINDOW WIPER – An alternative screen clear system.
CHARACTER EXTRACTOR – Borrow those nice character sets
MAZE GENERATOR – Create your own fun.
HIRES ANIMATOR – This difficult subject made easier
SPRITE DRIVER – Design your own platform games without fuss.
ROTATRON – Demonstration of sprites and sounds.
TEXT COMPRESSION – How to squeeze a gallon into a pint
SCREENS Make up your own help screens and keep them in memory
INTERRUPT POINTERS – Geos style windows and pointers for you.

All orders should be sent to. **SELECT SUBSCRIPTIONS LTD.** 5, River Park Estate, Billett Lane, Berkahmsted, Herts, HP4 1HL.
Please allow 28 days for delivery



ONE DAY AT SCHOOL...
OH NO! NOT MATHS AGAIN!
MATHS TEXTBOOK
AND MR JONES IS SUCH A ROTTER!!
YOU KNOW YOU TOLD ME ABOUT THE SCHOOL COMPUTER CLUB?
YEAH?
MATHS TEXTBOOK
WELL I RANG F.A.S.T. AND TOLD THEM ABOUT IT.
AND HOW MR JONES LETS EVERYBODY COPY SOFTWARE, AND EVEN DOES IT HIMSELF?
I'M NOT SURE WHAT THEY'LL DO, BUT THEY WERE INTERESTED
I HOPE THEY GET HIM
THEN I'LL SHARE THE REWARD WITH YOU!!

£1000 REWARD FOR INFORMATION LEADING TO A PROSECUTION & CONVICTION

THIS CAMPAIGN IS ORGANISED BY ELSPA

EUROPEAN LEISURE SOFTWARE PUBLISHERS ASSOCIATION

***ANY* INFORMATION ON PIRACY SHOULD BE PASSED TO F.A.S.T. (THE FEDERATION AGAINST SOFTWARE THEFT)**

TELEPHONE 071-497 8973

PIRACY IS THEFT

Jason Finch is the man on the spot

Adventure Helpline

After being inundated with requests we start a new regular feature aimed at the dedicated Adventurists

Welcome to Adventure Help. This is a new feature of Commodore Disk User that will be running for a few months to provide you with hints and tips for the completion of some of the adventures that have appeared in CDU in the past. Approximately five issues will be dedicated to each adventure, first of all taking you through the game with the aid of general hints (if you can't see them, I assure you, read the sentences again and they are likely to jump out and stare you in the face!), followed by the production of a map and complete list of the vocabulary and finally a detailed set of location solutions that will enable you to finish the entire game. The first adventure that will be covered is Kron, written by Tony Rome, which was published in the December 1989 issue of CDU. In this first article I shall cover the first stage of the adventure – the Sea of Storms and how to negotiate it, collecting what is necessary for later success.

When you have found the boat on the beach you will be able to sail out on to the open sea. Here, though, you face a problem, namely that the boat drifts aimlessly when you try to go anywhere. However, this is hardly surprising in a small wooden boat when the sails are down! Don't waste too much time here, too long out at sea can lead to death from exposure! Travelling eastwards, and remembering not to discard your boat for fear that you will drown, you will eventually arrive at the foot of a dark cliff. Upon finding the steps up the side of this cliff you can climb them to enter the Boran monastery. The clue for how to find the steps lies in the description of the location. Hang about a minute and the solution may reveal itself! (Sorry, couldn't resist giving a subtle hint!)

Once inside the monastery you should look around and obtain anything you can. That is the essence of adventuring – keep on examining everything in sight and you are bound to find something, even if it is only a scroll! Just remember when you go back down the cliff you will have to get in your boat again, otherwise it is bye-bye adventurer. Upon leaving Sark there are four possible destinations, excluding the beach, that you should visit. Of course the other outcome is death – travelling towards rushing water will only result in you being sucked into somewhere you don't want to be!

The four possible outcomes all result in you arriving on dry land. An island lies in the middle of the sea (not surprising since this is the usual location for an island!) and this is the first place you should visit. It will be beneficial for you to enter the cave – there is something in there that you want, I assure you. By the way, the twigs aren't just there to conceal the entrance! (You'll have to wait until next month to find out why!)

The second possible destination is to the far west of the Sea of Storms – southwest (ish) of the beach. You are told that there is a ledge a few feet above you which, obviously, you must climb onto. However, how are you going to get up there? Perhaps the tree could provide some help and what was that you found in the cave? Also, don't forget the tree is old and withered and might break if you are carrying anything heavy. When you have managed to get onto the ledge, and I have given you enough hints!, you will hear the source of the cries that you heard from below. An eagle is trapped, so do the courteous thing and you may get rewarded. I won't say exactly what else you must have, but if you have got to the caves further on in the adventure, remember that they are dark and you may need some form of light.

The third place you should go is immediately southeast of where you have just been. You hear voices that whisper your name… or is it just the sound of the wind hurrying across the sea? Well have a listen and you may find out what it really is. Should you for any reason discover a clam then think hard about how you are going to prise it open.

Now to the last location, the north shore of Sark, north of the great Caves of Goth. The first stage is the tunnel that carries on deep into the caves. Don't forget that it will be dark in there. Until next month, have fun!

Aleatory Music

A new approach to composing music.
Presented by Vic Berry.

There are two programs on the disk "ALLEATORY MUSIC" and "MACRO EDITOR". Both of these programs are fully compatible with "SID SEQUENCER", published in CDU May/June 89 issue. ALEATORY MUSIC can be used independently MACRO EDITOR requires music and sound files created by either ALEATORY MUSIC or SID SEQUENCER

The term Aleatory Music could be defined as music created with an element of indeterminacy. For instance, it is rumoured that Mozart used a dice to determine the musical structure of one of his piano sonatas. Modern composers such as Xenakis and Cage have used more complex systems to provide structures, pitches and durations. Cage used the "I Ching", a sort of oriental tarot, where characters are translated from sticks which are thrown onto the ground Xenakis uses computers to calculate the movement of individual instruments with equations such as Poisson's law, a formula which is used to model the appearances of rare events such as the decay of radioactive materials. In music, functions or probability tables can be used to determine structures, pitches or durations. These functions or probability tables form the raw materials that can be combined, controlled and moulded into any musical structure

The ALEATORY MUSIC PROGRAM allows the user to set up their own probability tables to define pitch and octave for each point in a music score The degree to which a final pitch can be predicted depends on the probability tables the user defines. The more unpredictable the expected output from the program the more chaotic or random the music will sound. The more the user limits the choices made by the computer the more regular or ordered the output will be Order can be imposed until the system creates a monotonous or predictable result. In music a compromise or balance between the opposing elements of chaos and order produce the best results. By limiting the possible outcome of each note it is possible to create a musical structure from random numbers.

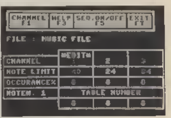
Imagine the score you will be creating as a grid, three vertical boxes for the three SID chip voices, and as many boxes horizontally as there are notes to be in the final piece At the beginning all the channels are clear ie. rests. The user selects the number of notes to be stored into the grid. When the COMPOSER is operating, the program stores a defined number of notes into the score. Each note is then stored individually by first taking two random numbers to select a particular note number and channel The note number and channel number is then used by the computer to point to another grid identical to the score grid. This grid stores a number which represents the number of the probability table from which the final note is determined. The program user defines which tables are to be used for each voice and note number The program has a maximum of 16 user defined probability tables which contain the probabilities of a particular octave and note letter name occurring. The computer then uses another two random numbers to determine the octave and note letter name which is finally stored into the score grid. The selected note is stored at the previously calculated note number and channel.

Program Operation

To start ALEATORY MUSIC load and run as a normal basic program. At the start the program loads three machine code files; the machine code music sequencer, the note table and the default probability tables. After the files are loaded the user is presented with the MAIN MENU. Options are selected by moving the cursor and then pressing the RETURN key. Each option will be dealt with in turn.

Edit Definitions

The screen will show the number of notes in the score for each channel and the percentage (%) chance of a particular channel being selected when notes are stored into the music score. The current filename for the score is also shown which can be changed by pressing the 'F' key The note number on the left side of the screen shows which point of the score the editor is looking at. The numbers



```
CHANNEL | HELP | SEQ.ON/OFF | EXIT
  F1    |  F3  |     F5     |  F7

FILE : MUSIC FILE

             | [illegible] |    |
CHANNEL      |      1      | 2  |  3
NOTE LIMIT   |     40      | 24 | 64
OCCURANCE%   |      0      | 0  |  0
NOTEN. 1     |     TABLE NUMBER
             |      0      | 0  |  0
```

under each channel box represent the probability table number which is used at a particular point in the score The table number is used by the COMPOSER to determine the actual note to be stored into the music.

The channel edit cursor is moved by pressing the 'f1' key To change the number of notes held in the edit channel hold down the 'CTRL' key and press the 'L' key. Try setting up a short phrase by setting the following limits on the score.

CHANNEL 1 – 32 Notes
CHANNEL 2 – 16 Notes
CHANNEL 3 – 16 Notes

Note with this particular configuration channel 1 takes twice as long to play through the score compared to channels 2 & 3 This creates the effect of having two voices providing a short repeating accompanying phrase with one voice providing some variation This can be a useful technique to use in your own music.

Pressing the O key will increment the probability of a particular channel being selected (Pressing 'SHIFT' & 'O' will decrement the channel percentage). The default increment is set at 5%, but this can be changed by pressing the 'P' key. For our test score set up the following percentages:

CHANNEL 1 – 50%
CHANNEL 2 – 25%
CHANNEL 3 – 25%

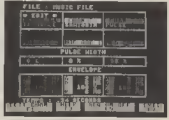
Note that the percentage chance of channel 1 being selected is twice that of the other two channels, because the number of notes in channel 1 is twice that of the other channels. This should ensure that the density of notes in each channel is roughly equal. It is not necessary to do this in your own scores, having different densities in each channel can be used to create different effects. However the total percentages for each channel must add up to 100% for the COMPOSER to work.

Now that the basic framework of the score has been set up, each note in each channel must reference a probability table. Pressing the 'N' or 'SHIFT' & 'N' keys the editor will move through the score. Pressing the '+' or '–' keys increases or decreases the probability table number for a particular point of the score in the edit channel. There are a maximum of 16 tables to be accessed. At the start the program sets all channels and points in the score to reference probability table 1. This will do for the purposes of the demonstration, but extremely complex structures can be set up with the editor. A screen is provided to help the user remember all the functions, this is obtained by pressing the 'f3' key. Pressing the 'f7' key will exit back to the MAIN MENU.

Edit Tables

This screen shows the individual probability tables, by pressing the 'f1' or 'f2' keys each table can be accessed. The top part of the table shows the twelve note letter names (marked NTS) and their respective percentage chances of being selected when the table is accessed by the COMPOSER. The note letter name cursor is controlled by the cursor left & right keys. The percentage for each note letter name is increased or decreased by the '+' & '–' keys. The default increment is 5%, but it can be changed by pressing the 'P' key. To continue with the example enter the following note letter name percentages into probability table 1:

C – 15%
D – 5%
E – 10%
G – 10%
A# – 10%



Notice that as a note letter name percentage is increased the percentage chance of a rest occurring automatically decreases. This is shown in the box marked 'RT%'. Musicians will know that the note letter names selected in the above table will make up a ninth chord of C. I chose these values so the example would yield a tonal sound, however any values can be used.

The lower part of the table shows the octaves (marked 8VE) and the percentage chance of occurrence when the table is accessed by the COMPOSER. The octave cursor is controlled by the up & down cursor keys and the percentages increased or decreased by holding down the 'SHIFT' key and pressing the '+ & – keys. For the example set up the following percentages.

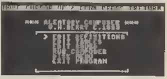
OCTAVE 2 – 10%
OCTAVE 3 – 25%
OCTAVE 4 – 30%
OCTAVE 5 – 25%
OCTAVE 6 – 10%

Note that the octave percentages must add up to 100% for the COMPOSER to work. In the example the octave percentages show that extreme high and low notes should occur less frequently than middle frequency notes.

In the example (or default setting) score all channels are set to probability table 1, but you can define up to 16 tables if it is required. All functions are detailed on the help screen 'f3'. To exit the editor press 'f7'.

Run Composer

The COMPOSER is active as soon as this option is selected. The screen shows the note selected through the probability tables and the note it will replace in the memory. A single note can be stored every time the 'RETURN' key is pressed. Pressing any other key the note will not be stored and another note will be selected. However 100 notes can be stored at once by pressing the 'f1' key. The number of notes to be stored can be changed by pressing the 'S' key and entering another value.



If you have followed the example so far you will be ready to store some music, press 'f5' to turn on the machine code sequencer so you will be able to hear the notes build up as they are stored. Press 'f1' to start storing notes. Pressing 'f3' will show the help screen which details all the commands accessed from the composer screen. 'f7' will exit the composer.

Edit Sound

Once a short musical phrase has been created you are ready to alter and experiment with the actual sound of each voice. The SID editor is identical to previously published SID SEQUENCER program. The user has control over the envelope, waveform, and modulation of all 3 channels. The tempo or speed of the music can also be adjusted from this screen. All the controls are detailed in the help screen called up by pressing 'f3'.

Creating Music from Short Phrases

Musical phrases that bear a close resemblance can help give a piece of music coherence or unity. Musical phrases which are not similar provide variation and contrast in music

In the above example a short musical phrase can be generated from a simple table Several phrases can be created and saved onto a disk from one set of probability tables Combining several music files generated from one set of tables can create subtle variations. By continuing to store notes into the score which has already been saved it is possible to increase the similarity between music files By creating a tree of related files it is possible to create differing degrees of similarity between short phrases The relationships between short phrases are the building blocks to be used in creating your final composition MACRO EDITOR can combine these building blocks of music much like a jigsaw puzzle

A Tree of Related Music Files

If a fixed number of notes are stored into an empty score a first order file is created this action can be repeated any number of times. Superimposing a further number of notes onto a first order score produces a second order file This sequence of notes is not only related to the probability tables used, but is related to the residual notes left in the score If the number of notes stored into the score is small in comparison to the size of the score there is a high degree of similarity between the music files. The process of creating different orders relating to one set of probability tables is limitless A tree structure of results could be obtained as shown below.

All the numbers represent a music file

```
    [illegible]r
 [illegible]rder
     nd
        3rd
            4th Order

 [illegible]  1 1  → 1 1.1 → 1 1 1.1 [illegible]
                   → 1 1 2
                   → 1 1 3
```

```
   [illegible]   [illegible]
                 [illegible]
   [illegible]   [illegible]
                 [illegible]
                 [illegible] 3 3
                  ...
   [illegible]
                 2 1 1
                 2.1.2
                 2.1 3

                 2 2 1
                 2.2 2
                 2.2 3
               etc.
```

If there is not sufficient contrast in the phrases from a tree of files created by one set of probability tables the user can use several sets to create short musical blocks. On the CDU disk I have given a number of examples of both probability tables and music scores A tree structure was used in compiling the ATONAL MUSIC file. To create a tree the number of notes stored for each order was 36 Files were selected and combined using the MACRO EDITOR as follows.

```
BLOCK A =
[illegible].1 1)+(3 1.2)+(3.1 3)+(3 [illegible]
[illegible]=60 notes)
[illegible]LOCK B =
[illegible] 1)+(1 3 2)+(1 1 3)+(1 3.2)+[illegible]
[illegible]=60 notes)
[illegible]LOCK C =
([illegible].1)+(2 2 3)+(2 3 1)+(2 3 2)+(3 [illegible]
([illegible]=60 notes)
[illegible]INAL SCORE = (B*2)+A+C
([illegible]=240 notes)
```

Macro editor operation

Load and run MACRO EDITOR as a normal basic program if the machine code sequencer and note table are not already present in the computer's memory there will be a short pause before the MAIN MENU is shown. The options from the MAIN MENU are selected as before using the cursor and pressing the 'RETURN' key. There are three main options, the editor itself, a hardcopy facility, and the usual disk handling functions

The editor

The EDITOR is similar in layout as the NOTE EDITOR in the SID SEQUENCER program. The screen shows a page of music (64 notes) belonging to one channel The highlighted note data in the top left corner of the grid shows the position of the cursor. All the functions available from the EDITOR are shown by calling up the HELP function 'f3' All functions work on each channel individually, this provides greater flexibility in designing your score. The main functions are, deleting a marked block of music ('CTRL' & 'D'), overwriting a block of music ('CTRL' & 'W'), and inserting a block of music ('CTRL' & 'I') Before calling any of the EDITOR functions you must mark the beginning and end of the block of music you wish to delete or copy (move the cursor to the point you wish to mark and hold down the 'CTRL key and press 'B' or 'E'). When overwriting or inserting music mark the block of music and then move the cursor to the point you wish to insert or overwrite before pressing the function

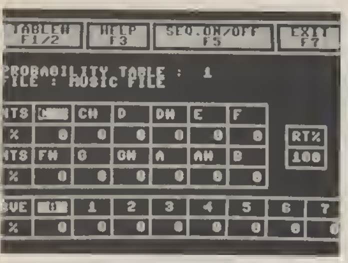
Block errors will occur if the END block marker is less than the BEGINNING marker and you are not allowed to overwrite or insert within the marked block. The remedy to this problem is to copy your block in two stages, however there is a limit of 255 notes stored in each channel

Hardcopy facility

This option provides a hardcopy of the score in two formats, it is generally useful in helping the user visualise the score during editing. The horizontal format prints each channel in turn, the output looking similar to the EDITOR layout The vertical format shows the three channels together This is useful if you want to find out how the parts interact or which notes are sounding together Make sure your printer is switched on when selecting this option or the program will crash.

Disk menu

All the functions are straight forward loading, saving and scratching files. The addition of an APPEND MUSIC allows the user to join files together,

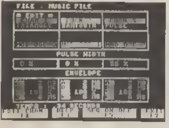


```
TABLE#   HELP   SEQ.ON/OFF   EXIT
 F1/2     F3        F5        F7

PROBABILITY TABLE :  1
FILE : MUSIC FILE

NTS  C    C#   D    D#   E    F
%     0    0    0    0    0    0      RT%
NTS  F#   G    G#   A    A#   B       100
%     0    0    0    0    0    0

VUE  0    1    2    3    4    5    6    7
%     0    0    0    0    0    0    0    0
```

however the program does not check how much room there is available for music data so you must be careful. the limit for each channel is 255 notes In practice the best results are achieved by keeping your music files quite short to give greater flexibility in manipulation by the EDITOR

Demo files

On the disk there are several music, sound and probability files for you to use and experiment with



```
FILE : MUSIC FILE
= EDIT =
TRIANGLE      SAWTOOTH      PULSE
                 PULSE WIDTH
0 %           0 %           50 %
                  ENVELOPE
TEMPO :  24 SECONDS
EDIT CHAN   HELP   SEQ ON/OFF   EXIT
```

Progression

The probability file was designed from a simple 9 note harmony exercise The exercise (from Schoenberg's Theory of Harmony page 172, ex.118a) was reduced to six chords. The six chords were then programmed as six probability tables Each note of the chord was given an equal percentage chance of occurring unless the note was doubled (ie a note letter name occurring in two voices) and then the percentage chance of the note occurring would be doubled The score was compiled using several saved short phases which were joined together using the MACRO EDITOR. In the final score you will notice the number of notes in each channel are different, this makes the individual channels drift out of phase with each other. The phasing is spaced in multiples of 9 notes apart so that the music continues to harmonise producing the same chords each time with different melodies

CHANNEL 1 – 108 Notes
CHANNEL 2 – 72 Notes
CHANNEL 3 – 90 Notes

Atonal music

Described above in detail Each note letter name has an equal chance of occurring in the probability tables. The music lacks harmony or any feeling of tonality

In C

Here all the notes of the C major scale (All notes without a '#') were given an equal probability of occurring A related tree structure as described above was used to produce short phrases Some of the music files were joined together to produce a gradual effect, each voice moving to the next file individually as shown below.

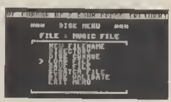
CHANNEL 1 · file A – file B – file B – file B
CHANNEL 2 · file A – file A – file B – file B
CHANNEL 3 · file A – file A – file A – file B

Sigma

The probability tables were calculated using a purely theoretical structure devised from the numbers 5 & 3. The interference pattern created by superimposing these two unit values is:

[illegible] = 3+2+1+3+1+2+3

Truncating this sequence and superimposing the sequence over the chromatic scale (ie all note letter names)



```
*** DISK MENU ***
FILE : MUSIC FILE
   NEW FILENAME
   DIRECTORY
   DISK COMMAND
 > LOAD FILE
   SAVE FILE
   SCRATCH FILE
   EXIT VALIDATE
   EXIT MENU
```

made it possible to obtain the following musical scale as one possibility·

C D# F F# A A#

These notes were then given different chances of occurring in the probability tables. The points at which certain probability tables would be accessed by the COMPOSER were also decided to an identical structure devised from the numbers 5 & 3. This creates an element of harmonic rhythmic movement. The results from several files were then taken to complete the score

Any method or technique can be used to create music with the program. The process of collecting batches of music files before making a subjective decision about their final use is the technique employed for the demo files' This is just one method from many you can devise for yourself I think you will agree the results from experimenting with this program are astonishing

Sprite BASIC

A basic extension that makes sprite maintenance relatively easy **by Gordon Moyes**

When you were buying your '64, you probably looked at the box. Great synthesised sound, sprites and graphics were all promised, but they left one line out: 'Worst Basic of all time'. This program gives you control over 32 sprites as well as sound and various other commands.

How It Works

The various commands basically replace a series of pokes. The sprite routines also control a raster interrupt, with a maximum of four zones. Each zone has a separate set of sprite pointers set aside for it in RAM. At the beginning of zone 0, the 0 zone sprite data is loaded into the vic chip. When zone 1 is reached, zone 1 sprite data is loaded into the video controller etc.

The restriction is that a zone 0 sprite can only be at the top of the screen, whereas a zone 3 sprite can only be at the bottom. Sprites can, however be tagged together to allow them to cross the various zones, at a cost of the usage of a sprite in the higher numbered zone.

The raster routine also takes care of the duration of the play command. The duration is decreased at the bottom of every screen raster and when it reaches 0 the sound is determined.

None of the sprite commands will work and the play command will play forever (or until a release command) unless at least one raster zone is activated.

Sprite Commands

SPRXY group, number, xpos, ypos
This command positions the relevant sprite at position xpos, ypos. Groups should be between 0 and 3 and numbers should be between 0 and 7 (4 groups * 8 sprites = 32 sprites). Xpos should be between 0 and 511 and ypos between 0 and 255.

SPRAT group, number, priority, expand x, expand y, multi, own colour
This command sets a sprites attributes. Priority should be 0 if you want sprites in front of background text and 1 if not. Expand x and y should be 1 if the sprites is expanded and a zero otherwise. Multi is 0 if you want a hires sprite and 1 if you want a multicolour sprite. Own colour is a number between 0 and 15.

SPRSLOT group, number, slot
Set a sprites slot (picture source) number. Slot is a number between 0 and 255. Note that sprite slots 64 to 127 are not usable (the video chip sees character data here), and any slot less than 32 will corrupt screen/system data.

SPRITE group, number, display
Display a sprite. Display should be a 1 to turn a sprite on or a 0 to turn it off.

SPRMULT colour1, colour2
Set sprite common colours (for multicolour sprites). Colour 1 corresponds to location 53285 (bit pair 01) and colour 2 to location 53286 (bit pair 11).

SPRTAG number, group1, group2, group3
Tag sprites of the same number together to allow them to cross a sprite zone. Group should be 0 to untag and 1 to tag to the sprite in the group 1 less that the tagged group. Note that group zero sprites are always untagged. e.g. SPRTAG 0,1,1,1 would mean that any command issued for sprite 0,0 would also be followed by 1,0,2,0 and 3,0 so that it would appear that sprite 0,0 can go anywhere on the screen. SPRTAG 5,1,0,1 would combine sprite 0,5 with sprite 1,5 and sprite 2,5 with sprite 3,5. This command only affects sprite commands issued after sprtag!

MAXZN zones
Define the number of sprite zones to use. You must do a maxzn 0 (turn off) before any disk/tape i/o. Zones should be between 0 (off) and 4.

RASTER raster 1, raster 2, raster 3
Change the sprite zone positions. The raster positions must be greater than 50 and less than 240. They must also be in increasing order and at least 10 apart, otherwise flickering of all sprites will result and the normal interrupt (keyboard etc) will run a lot slower. (It wraps around to the top of the screen), and also takes care of the keyboard interrupt and play duration.

Note that at least 10 raster lines must be left between the raster position and the minimum y position of that sprite zone. The y position of a sprite should not cause the bottom of that sprite to cross a sprite zone, or it may disappear, flicker, reappear in another position completely or be duplicated!

SPRSPR (group, number) = variable
Is a sprite colliding with another sprite? This command will make the variable 0 if the sprite is not colliding and a one if it is. Note that it does not specify which sprite it is colliding with. The variable MUST be numeric (numerical arrays are ok too). The collision register for the specified sprite only is cleared once read (unless the sprite is still colliding).

SPRBAK (group, number) = variable
Is a sprite colliding with background information? The format is the same as SPRSPR. Note that background multicolour bit pair 01 is considered transparent for collisions, as well as the background colour.

Sound Commands

VOLUME level
Assign the overall volume level. Level should be between 0 (off) and 15 (loudest).

ENVELOPE voice, attack, sustain, decay, release
Set the adsr envelope for a particular voice. Voice should be between 0 and 2. Attack, decay and release are all times and should be between 0 (fastest) and 15 (slowest). Sustain is a volume level and should also be between 0 and 15.

WAVE voice, waveform, pulse cycle, sfx
Set the waveform, pulse duty cycle and any special effects. Waveform should be 1 = triangle, 2 = sawtooth, 4 = pulse, 8 = noise. You can try mixing them together (adding the values) but it is not recommended

Pulse cycle is the duty cycle of the pulse waveform (set this to 0 unless you are using the pulse waveform) and should be between 0 and 4095

PLAYING (voice) = variable
This will make the vanable 0 if no sound is being played on that voice, and a one if there is.

Other Commands

CLS	–	Clear the screen
HOME	–	Put the cursor at (0,)
AT (x,y)	–	Put the cursor at x,y
BRDR colour	–	Change the border colour
COLOUR colour	–	Change the text (ink) colour
TEXT mode	–	0 = hires, 1 = multicolour
CMULT col1, col2	–	Set text multicolour registers
CSET set number	–	Character set (2=upper, 3 lower)
QUIT	–	Reset the computer

JOY (port) = vanable
Port should be either 1 or 2.
Vanable will be 1 = up, 2 = down, 4 = left, 8 = right or possibly a combination

BTN (port) = variable
Vanable will be a 21 if the fire button was pressed and a zero if not.

Sfx should be 0 = normal, 1 = synchronise, 2 = ring modulate, 3 = ring mod and sync and 4 is disable oscillator

FILTER cutoff, resonance, type, v0, v1, v2
Set filter vanables. Cutoff is a cutoff frequency between 0 and 2047. Resonance is between 0 and 25 Filter type is 0 = no filter, 1 = lowpass, 2 = bandpass, 4 = highpass, 5 = notch reject. V0, v1 and v2 should be 0 if you do not want the corresponding voice to be filtered and a 1 if you do

CLEARSID
This command clears all of the sid registers to 0, turning all sound off It has no parameters.

PLAY voice, frequency, duration
Make a sound or noise The frequency is a value between 0 and 65535 and notes can be found in the user manual on page 152. If you multiply the high frequency given by 256 and add the low frequency, you end up with the final frequency. The duration is a number between 0 and 65535 If it is 0, the note will play until a release statement is issued. The duration is in 1/50ths of a second. i e PLAY 0, 8770, 50 would play an octave 5 note C for 1 second Your program is not held up while the duration is counted down.

RELEASE voice
This command will stop sound that is being played It can be used to halt a sound prematurely or to terminate a sound created with a play duration of 0

Using Sprite Basic

I would recommend that you use cset 1 at 2048-4095 (set 1 on the 3 in 1 editor by Tony Crowther – published in Your Commodore) if you are using user definable characters, leaving sprite slots 128 to 255 free for sprites

If you want to move the start of basic up to 16348 (to protect it from graphics data) you type poke 44 64: poke 16384,0:new

Machine Language

All of the sprite ad sound commands can be used directly from machine language JSR calls, without requiring the basic section of the code The calls are.

SPRXY – jsr 49746
49159 = group, 49148 = number, 49155 = xpos lsb, 49156 = xpos msb, 49157 = ypos
SPRAT – jsr 49659
49159 = group, 49158 = number, 49160 = priority 49155 = xexp, 49157 = yexp, 49161 = multi, 49162 = own colour
SPRSLOT – jsr 49799
49159 = group, 49158 = number, 49160 = slot
SPRITE – jsr 49725
49159 = group, 49158 = number, 49160 = display
SPRMULT – jsr 49576
49160 = colour 1, 49161 = colour 2
SPRTAG – jsr 49589
49158 = number, 49160 = grp1, 49161 = grp2, 49162 = grp3
MAXZN – jsr 49814
49155 = zones
RASTER – jsr 49557
49160 = raster1, 49161 = raster2, 49162 = raster3
SPRSPR – jsr 49175, result in 49160
49159 = group, 49158 = number
SPRBAK – jsr 49209, result in 49160
49159 = group, 49158 = number
VOLUME – jsr 49298
49160 = volume
ENVELOPE – jsr 49313
49158 = voice, 49160 = attack, 49161 = decay, 49162 = sustain, 49163 = release
WAVE – jsr 49364
49158 = voice, 49160 = wave, 49161 = pulse lsb, 49162 = pulse msb, 49163 = sfx
FILTER jsr 49478
49160 = cutoff lsb, 49161 = cutoff msb, 49162 = resonance, 49163 = type, 49155 = v0, 49156 = v1, 49157 = v2
CLEARSID – jsr 49255
PLAY – jsr 49412
49158 = voice, 49160 = frequency lsb, 49161 = frequency msb, 49162 = duration lsb, 49163 = duration msb
RELEASE – jsr 49453
49158 = voice
PLAYING – jsr 49243; result in 49160
49158 = voice

If you want to add your own interrupt control to the bottom of screen raster (to smoothly scroll the screen or to swap display pages etc.) you can vector it through locations 50225 and 6. The sequence 'sei.lda #>inthndle sta 50225.lda #>inthndle·sta 50226;cli' will achieve this if your interrupt routine starts at inthndle Your interrupt routine must finish with a JMP $EA31 or pla: tay:pla:tax pla rti if you don't want any keyboard input

If you are going to swap display pages, you will have to change the rasters sprite pointer addresses You should store the pointer high byte in 49933, 49957, 49981, 50005, 50029, 50053, 50077 and 50101 The pointer high byte is the video base address high byte + 3 i.e. if the screen is at 1024 (normal) then the pointer is 1024/256+3 = 7.

From Here

This utility should greatly enhance your machine language programming and add new life to basic Get writing and try it out!

Adventure Writing

Another new series gets under way in the shape of exploring the theories behind adventure writing

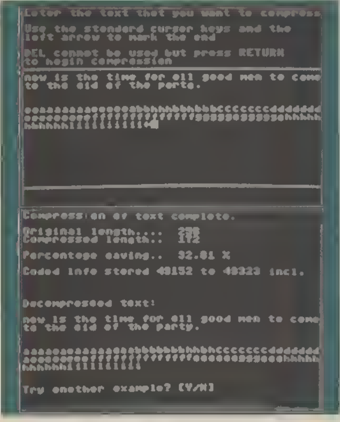
By Jason Finch

A number of software packages have been written in the past, for both tape and disk, to assist with the writing and development of adventure games. For example, the Graphic Adventure Creator and the Quill to name just two Each one has its own particular characteristics that make it unique and they undoubtedly simplify the process of writing such a game However, it is my opinion that the satisfaction obtained by writing your own adventure and also programming it yourself, ignoring the comparisons between quality, is much greater than that obtained by planning one and then loading up another program whereby you simply enter the text and roughly what should happen And with certain packages that help you to program adventures you have to credit the software company that originally published the program Of course, writing an adventure like that is not cheating and gives a non-programmer a great deal of satisfaction when complete However, this particular series is directed towards people who feel confident with the BASIC language

This series of articles will not only familiarise you with the main aspects of adventure games, it will also carefully look at and consider all attributes that could be taken. In a later issue I shall introduce you to a short and quite simple example adventure that I have written, with graphics – a subject that will come under the spotlight next issue This will be built up issue by issue until you finally have a complete working graphical adventure. Of course, you will know how to complete most of it as I shall be giving full explanations of each months piece. At the end I shall provide a program that will link everything together and save it onto your own disk

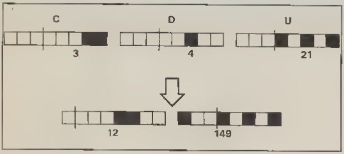
However, in this, the first of the new series, I shall delve into possibly the most important part of an adventure – the text First, though, let's make a couple of things clear. The type of adventure covered by this series is one where the player types in commands which are then executed and the outcomes displayed on the screen There may well be other characters involved but I am not going to talk about arcade adventures or role-playing games

Without doubt, every adventure that is written contains a great deal of text, whether it be regarding the vast amounts used to describe the locations or whether it be the 'vocabulary' of the computer The latter cannot really be condensed in any way but the former, the room descriptions, can. So, this month I am going to explain quite an important, although not absolutely necessary, part of writing an adventure Most location descriptions will consist of, usually, lower case letters with the odd capital here and there, like most pages of this magazine If you are able to somehow compress the text easily, and recall it with the same ease, then it may be profitable to do just that On the disk you will find a program, written in BASIC, called "AW-COMPRESS" Eventually I shall convert the 'decompression' part to machine code for speed but I have left it in BASIC so you can see what is happening



```
Enter the text that you want to compress

Use the standard cursor keys and the
left arrow to mark the end

DEL cannot be used but press RETURN
to begin compression

now is the time for all good men to come
to the aid of the party.

aaaaaaaaaaaaaaaabbbbbbbbbbbbcccccccddddddd
eeeeeeeeeffffffffffffffffgggggggggggghhhhh
hhhhhhiiiiiiiiiii

Compression of text complete.

Original length....:  256
Compressed length..:  172

Percentage saving..   32.81 %

Coded info stored 48152 to 48323 incl.


Decompressed text:

now is the time for all good men to come
to the aid of the party.

aaaaaaaaaaaaaaaabbbbbbbbbbbbcccccccddddddd
eeeeeeeeeffffffffffffffffgggggggggggghhhhh
hhhhhhiiiiiiiiiii

Try another example? [Y/N]
```

The method is only one of many and has the ability to compress text to



up to just two thirds of what it originally was, although the average sentence will compress to about three quarters of its usual length. The secret lies in taking each set of three characters in turn. Each letter is considered separately and then converted to a number between zero and thirty-one. The letters A to Z are represented by the numbers one to twenty-six inclusive, the space by zero and then the most common punctuation marks, such as the full stop, comma, question and exclamation marks are converted to twenty-seven onwards. This means that each usual eight bit number that represents the character is condensed to just five bits. Three five bit numbers, when stringed together, form one fifteen bit number or near enough two eight bit numbers with one bit spare. Therefore, the usual three numbers that represented the characters have been coded and become two numbers. Of course, if you require a capital, or some other character then set that spare bit to a one and make the second number in the pair the actual POKE, or more commonly the ASCII code for that character. In case you didn't understand all that I have represented it in the figure, you will see the original three bytes – the letters C, D and U. The boxes beneath each represent the binary form of the POKE code. Shaded boxes represent a one. The necessary bits are chopped off and then everything is strung together below. You see the far left hand bit is clear. If that left hand byte is made the value 255 then the computer can be forced to read the second number as the actual ASCII or POKE code as I mentioned before. But what happens when the second or third byte of a sequence requires special attention? A value is assigned that instructs the computer to leave that pair of bytes and go to the next.

Have a look at the program on the disk and see exactly how it works. The concept is quite hard to explain although the theory is very easy. You will also need a decompression routine to decode the numbers into legible text. This is just one method, there are a few others. For example common words such as 'the', 'and', 'you' and so on could be given specific coded numbers that instruct the computer to print that word when the number is read. This would not have the same compression power but is another alternative. As I say, have a play about with the BASIC routine – it has got plenty of REM comments to take you through it step by step.

Screen display

You may relate the idea of having an excellent screen display to fast action arcade games but it is just as essential, if not more so, with an adventure. A person is likely to be sat at the keyboard for some while playing your adventure and so the screen display does not want to be dull and boring. Aspects such as colour and graphics can be brought in and further on I shall discuss the implementation of both. However, first you need to decide how your main screen will be split up or formed.

The first question you should ask yourself is: "Will my adventure have any pictures and if so, how much importance am I going to put on them?" If you decide to have an all text adventure then you will need to put as much thought into the screen display as you will if you are having a mixture of text and graphics, and if you decide to have pictures will they be visible all the time or optionally visible with the press of a key and what size will they be – whole screen pictures, about half a screen or perhaps you only want them to be very small so how about using eight multi-colour sprites? Some people believe that an adventure should be all text because the player can then conjure up in his own mind what the location should look like, whilst other people think that a screen full of text looks 'boring'. So how do you decide? In my opinion you should compromise and have it half text and half graphics with the option of having the graphics turned off if you don't want them. First let's consider a possible screen display for an all text adventure.

You could have a division about three quarters of the way down the screen with the top part being the "adventure window" where all the location descriptions and the messages are displayed, with the bottom few lines being the "input window", devoted to the inputting of the commands. The best approach, though, would be to have an integrated display where it is all one. The location is described with the prompt to type a command immediately below it. When you enter the command the whole display shifts up and the result of your action is shown and so on. In your adventure you will want to display the exits that are available from one location. You could either merge these into the text, for example – To the east a long pathway stretches over the hill, or you may want to have a separate line that tells you simply what directions are available – You can go north, east and southwest. The same sort of decision needs to be made about how you describe the objects that are visible in that location.

On the other hand, if you decide to have bitmapped graphics pictures you must decide how you are going to go about displaying them. Will they be full screen pictures loaded from disk as required or will they be drawn with a set of line, circle and box commands? The latter occupies less memory but it is not as easy to obtain satisfactory effects and is more difficult to program. If you have knowledge of machine code then you could write a routine to introduce "raster splits". This will allow you to have the top section of the screen displaying graphics whilst the bottom

Adventure Writing

is the text window. If you decide on this option, how large are the pictures going to be? I mentioned earlier the possibility of using multicolour sprites. This will give you a very small picture that can be displayed constantly but the resolution and variety of colours with this technique is very poor and should only be used if the picture is meant to be a simple stimulus

Now I have provided you with a few ideas as to the layout of the screen Forget the pictures for a while and let's concentrate on the text. At the start I commented on the fact that you are likely to have the majority of the text in lower case letters – most people prefer this Don't fall into the trap of having everything displayed in upper case Try to imagine what this magazine would look like if every article was written in upper case letters. The next most important thing is the use of colours It helps a lot in arcade games to have variety but does it in adventures?

There are sixteen different colours at your disposal although you should never use more than about three Don't dazzle the player with a huge array of colours throughout the text It is the content of the text that he is more interested in. However, always ensure that the colour is visible without straining your eyes. For example don't display red text on a brown background Likewise don't make the text too bright I wouldn't want to sit for any length of time looking at bright white text on a black background Although it makes it visible, tone it down a bit to light green, cyan or something of that nature but keep it coloured, not grey. If you feel that you need to change the colours then keep all the descriptions and messages in one colour and possibly vary the colour in which the commands are typed or messages related to actions are displayed In contrast to the text, the graphics, if you decide to include them and in whatever forsm they take, should be as colourful as the situation warrants

That wraps up the theory for this month In the next issue I shall say a bit about storage of information and actually programming part of the example adventure

C64 AMIGA C128

ONLY £149

C64 DISK DRIVE

FREE GEOS

COMMODORE 1541 CII DISK DRIVE, SLIMLINE CASE, POWER SUPPLY UNIT, 5¼" DRIVE £3 50 + P&P
THE ONLY DRIVE 100% COMPATIBLE WITH THE C64/128

NEW

AMIGA 500

ONLY £379

FLIGHT OF FANTASY
★ F29 RETALIATOR ★ RAINBOW ISLAND
★ ESCAPE FROM THE PLANET OF THE ROBOT MONSTERS
★ DELUXE PAINT II £5 00 P&P

NEW

LIGHT GUN £36.50

C64 LIGHT GUN PLUS
GAMES AND UTILITIES
LIGHT FANTASTIC UPGRADE PACK

£3.50 P&P

PRINTERS

COMMODORE MPS1230	£159 00
COMMODORE MPS COLOUR	£219
SEIKOSHA SP 180V (C64)	£149 00

C64 LIGHT FANTASTIC

LIGHT FANTASTIC PACK INCLUDES - LIGHT GUN 3D GAMES, PAINT PACKAGE, DATA RECORDER, GAMES
£140 00 + £5.00 P&P

100% ERROR FREE

DISKS

10x3 5" DSDD	£9.50
10x5 25" DSDD	£4 50
50 DISK BOX 5 25"	£5 99
100 DISK BOX 5 25"	£6.99
80 DISK BOX 3 5"	£6 95

MOUSE HOLDER	£2.50
MOUSE MAT	£4.50

C64 DATA RECORDER	£24.50
'LOAD-IT' DATA REC	£35 00
C64 POWER SUPPLY	£14 95
C64 MOUSE, MOUSE MAT & HOLDER	£26.50
KONIX JOYSTICK	£11.50

CARRIAGE £1.50

C.M.S

CROFTON MICRO SUPPLIES
45 WHITBREAD ROAD
BROCKLEY LONDON SE4 2BD

01-469 3246

IF AN ADVERT IS WRONG, WHO PUTS IT RIGHT?

We do

The Advertising Standards Authority ensures advertisements meet with the strict Code of Advertising Practice So if you question an advertiser, they have to answer to us

To find out more about the ASA, please write to Advertising Standards Authority Dept X, Brook House Torrington Place, London WC1E 7HN

ASA

This space is donated in the interests of high standards in advertisements

The Astrodus Affair

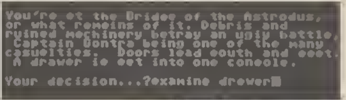
Can you survive the perils of the spaceship Astrodus and repair it to the full before flying off to a new life

By Mark Turner

It is 2120. Survival amidst the new Corporate Worlds System is becoming increasingly difficult, so when news reaches you that a technician is required aboard a nearby trading craft, you re first in the queue. After a cursory check from Captain Gontra, you're soon on your way to Custom Moon Salibus aboard the spacecraft Astrodus, enjoying your first regular job for over a year, and not even giving a second thought to the squalor of the work. After all, **they're** paying **you** for being alive.

For almost three weeks everything is great, eating, working, sleeping, simply enjoying life for the first time in an eternity (even tolerating continual abuse from that smarmy junior officer Stelen, although given half the chance you'll soon make him pay for his jocular comments) Then, true to the usual course of your life, your luck, and that of the crew's with it, takes a sudden nose-dive straight into an uncomprising abyss. Scavenger ships descend from nowhere onto the Astrodus, rocking the craft with straffing flak and panicking various crew members. Common sense tells you that every trading-craft must be prepared for such attacks, and within seconds the Astrodus' defence systems are operational, easily shielding the craft against the onslaught, whilst Gontra calmly informs everyone that "no diseased Scavengers are getting hold of this cargo, you mark my words'. Hence no-one is more surprised than the captain himself when, contrary to the predicted course of events, the Scavengers suddenly bring forth their secret weapon, a military-standard Damage Laser. Within seconds the Astrodus' defences are breached; a hole the size of small cargo-bay is opened in the outer surface, allowing the Scavengers swift and immediate boarding access They quickly began to scour the craft, laying waste to anything of no value at every turn, and you suddenly realise that there's to be no escape this time, for you're trapped inside an enclosed vessel with nowhere to run, and very little places to hide A sudden blow to the back of the head provides a solution to your dilemma, as you crumple to the floor, consciousness already floating away beyond reach towards a rippling, multicoloured horizon

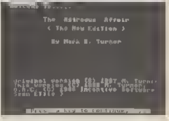
On regaining consciousness you find yourself staring through unfocussed eyes at the ceiling of the Bridge



```
You're at the Bridge of the Astrodus,
or what remains of it. Debris and
ruined machinery betray an ugly battle,
 Captain Gontra being one of the many
casualties. Doors lead south and east.
 A drawer is set into one console.

Your decision...?examine drawer█
```

Assuming you to be mortally wounded the Scavengers have simply left you to die in a pool of darkly-coloured blood, but for once your luck has held good, leaving you with only a large gash to the head and a splitting headache. At first, sitting-up sends the room spinning, but you rapidly recover your sense of balance, and eventually manage to stand on your own two feet. A cautious examination of the craft reveals a great deal of damage, and enough signs of violence to make your stomach heave, but you soon realise that everyone else has either been killed or taken, with most of the cargo having been stolen or destroyed As you slowly pull yourself together, the reality of the situation eventually dawns on you. You're alone, a trained technician aboard a damaged but repairable craft. The Astrodus is the key to a way of life you've always dreamed of, the realisation of all your childhood fantasies, and it's all simply waiting there, asking for the taking..

The Astrodus Affair is a foray into the original world of adventure gaming, as created by the likes of Scott Adams and Infocom, although whether it manages to attain such standards is your own opinion! (Probably not Infocom's!) On starting the game you're immediate thoughts will undoubtedly be concerned with simple survival, which will certainly be a bit tougher than the above commentary indicates However, your eventual aim is to repair and regain full working control of the Astrodus, and fly off into the sunset with the craft as your own. Standard adventure rules apply, ie N, S, E, U etc. for movement, with the usual LOAD, SAVE, SCORE, QUIT, GET,



```
[illegible]
        The Astrodus Affair
        ( The New Edition )

        By Mark E. Turner



 Original version (C) 1987 M. Turner
 This version (C) [illegible] M. Turner
 [illegible] (C) [illegible] Incentive Software
 Sean Ellis )

   Press a key to continue, [illegible]
```

DROP, INV, ENTER, TEXT, GRAPHICS, etc also available, as are many more (but no HELP command, just to be cruel!).

Since you could have been unconscious for days, don't be too surprised if some of the cargoes have begun to roam around the craft! In general, don't forget to examine EVERYTHING you come across, there are very few red herrings lying around, everything is logical (despite first appearances), and there are no major random elements in the game what-so-ever. As a final hint, there are at least five things to be repaired .

The Astrodus Affair was originally written in 1987 using the Graphic Adventure Creator, and then vastly changed and rewritten specially for Commodore Disk User in June 1989.

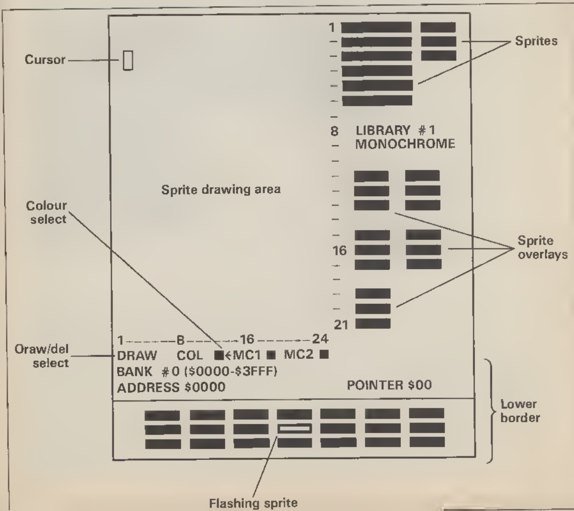
Sprite Generator 2

Yet another sprite creator joins the arsenal of graphics aids for the serious programmer

By Brian Graham

Sprite Generator is a utility for designing sprites for use in games, etc. It allows you to make monochrome or multicolour sprites, overlay sprites, animate them, store them anywhere in memory, save them and many other things

Screen display



Using the program

Once the program has loaded you will see a flashing cursor in the top left hand corner of the sprite drawing area This cursor is moved around the drawing area using either a joystick in port 2 or the cursor keys Pressing the fire button or RETURN will plot a point on the monochrome drawing area and a corresponding point on the two sprites In the top right hand corner of the screen Pressing SPACE will delete this point If you are using a joystick, you must press F3 to select DELETE mode, now pressing the fire button will delete rather than plot the point To return to the drawing mode press F1 (the current drawing mode is indicated under the bottom left of the drawing window)

Changing the colours

To select a different colour for the sprite just press F7 Other colours can be selected as shown below

COLOUR OPTIONS	KEY
Background	F5
Border	B
Sprite Colour	F7
Sprite Multicolour 1	,
Sprite Multicolour 2	.

The Sprite Colour is shown beside the letters 'COL' whilst Multicolours 1 & 2 are shown beside the letters MC1' & 'MC2' respectively, all of which are under the drawing area.

Colour modes

Sprites are capable of being drawn In one of two modes Monochrome or Multicolour. Each mode has its own advantages and disadvantages. For example, when in Monochrome mode, there is a higher resolution but you can only draw in a single colour (not necessarily black or white). This colour is selected by the Sprite Colour (F7). In the Multicolour mode you can use any of three colours specified by the Sprite Colour (F7) and the two Multicolours(, & respectively) but with a loss of half of the resolution. Having selected the two multicoloured sprites on the screen. To select the mode press M This toggles between the two modes When in multicolour mode pressing N will select which of the three colours is to be plotted The active colour is indicated by an arrow pointing towards a coloured box beneath the drawing area

Library sprites

To make it easier to create complex sprites there is a library of five sprites which can be independently designed and altered. These sprites can be positioned anywhere in an unmarked window at the right of the screen (press P to position the current sprite and use joystick or cursor keys and the fire button or RETURN to finish). Each sprite can have an independent colour mode, sprite colour and X or Y expansion (more about these later). They can also be combined to create one larger 'pseudo-sprite The five sprites can be selected by pressing the appropriate number, eg press 3 to select sprite number three The current sprite under construction Is shown on the sprite drawing area and also in the top right hand corner of the screen In this corner there are two sprites, one of which is permanently expanded in the X and Y directions whilst the other is variable (press C for X expansion and V for Y expansion). The current library sprite's number is shown above its colour mode near the top right of the screen

Sprite manipulation

The current sprite can be manipulated in various ways, for example, scrolling in all directions, mirroring and 'exclusive-or'ing (ie. toggling the set and unset bits) in Mono' mode only The controls for these are as follows

MANIPULATION OPTION	KEY
Scrolling up	U
Scrolling down	D
Scrolling left	L
Scrolling right	R
X mirroring	X
Y mirroring	Y
XORing	E
Clearing the drawingarea	Shift & CLR/HOM

Sprite interrogator

The Sprite Interrogator (or Sprint) is to give the user a complete view of all the sprites in memory. When loading the program all the sprites are merely random memory and appear as a bit of a mess However once a sprite has been created it can be stored anywhere (almost) in memory

The memory is split into four banks (0-3) selected sequentially by pressing O. The current bank is shown below the drawing area beside which is the range of memory locations that that bank contains (In hexadecimal notation). In the bottom border there are seven sprite frames, the middle one of which is flashing. To change their colours press ·' and to toggle their colour mode press ',. These frames show the sprites that are at a section of memory at any given time The frame to the left of the flashing one is the sprite immediately preceding the flashing one in memory while the one to the right is the one immediately after the flashing one. All the other frames precede or come after sequentially. To advance the address and thus examine the next sprite In memory press '+' and to lower the address press '–' The address of the flashing sprite is indicated underneath the bank indicator and its pointer is indicated to the right of it (both are in hex notation) For more information on pointers and hex please consult the Commodore 64 Programmers Reference Guide

If you want to animate, save or store sprites you must place them In a memory location

Storing the Sprite

To store the current sprite at the location of the flashing sprite press S. To take the flashing frame to the drawing area press G. To wipe the flashing sprite press W. To trade the current sprite with the flashing one press T. The best place to store sprites is in Bank 1 $4000–$7FFF) because it has nothing volatile in it, generally This will allow 256 sprites to be stored here It is suggested that novices always use this bank! NB NEVER store anything from $1000 to $3120 or you will crash the program.

Continued on page 44

65XX INTERFACING

We continue our excursion into the world of the 65xx family of microprocessors

By Steven Carrie

Now we will go on to look at some other facilities provided by the CIA. Let's pick up again with a look at the two interval timers, TIMER A and TIMER B. Their operational modes are controlled by two Control Registers, CRA for TIMER A (register $0E) and CRB for TIMER B (register $0F). Since each timer has 16 bits, they each require two registers, $04 and $05 for TIMER A, $06 and $07 for TIMER B.

The C64/128 operating system uses one of the timers from CIA 1 to cause the system interrupt that scans the keyboard. The timer runs in continuous mode and causes an IRQ signal about every 1/60th of a second. You can alter this by writing a new value to the TIMER A registers at $DC05, but doing so may have a strange effect on your computer's response to keypresses and the cursor flash rate.

The timers work by counting down from a predefined value ($0001 up to $FFFF) to zero at system (or external) clock rate. When the timer hits zero, a Timer Underflow event is signalled in the Interrupt Control Register (described earlier). Timers may operate in 'one-shot' mode; i.e. it counts down once and must be restarted manually; or they may run continuously; counting down to zero, and then automatically re-start.

The timer's count start value is set by storing the 16-bit value to the appropriate registers. This value will be stored until it is overwritten or the power is removed. Whenever a timer is started, this value is loaded and the countdown begins. If the timer registers are read, the current value of the timer is returned.

Each timer may be used to produce an output on a particular port bit. The nature of this output may also be controlled. Both timers may also be programmed to count pulses applied to certain port input pins. Timer B may also count Timer A underflow events, thereby allowing the timers to be used as one 32-bit timer/counter.

The two control registers CRA and CRB are used to define how the timers will operate. CRA controls TIMER A whilst CRB controls TIMER B. They also control the operation of the Serial Port system but we will leave this to one side for the moment.

The layout of the two Control registers is shown below.

Control Register A

bit→	7	6	5	4	3	2	1	0
	tod	spm	inm	load	run	out	pbon	start

start	Set this bit to 1 to start TIMER A or 0 to stop it. When in one-shot mode (see run mode) this bit is automatically set to 0 when underflow occurs.
pbon	By setting this bit to 1 you can cause TIMER A output to appear on pin PB6. When this happens, the line is forced

to an output regardless of the data direction register bit.

out — You can control the type of output on PB6 using this bit Output occurs when an underflow event occurs If this bit is 0, the output will PULSE for one cycle If it is 1, the output will TOGGLE, i.e. change logic level. This level will initially be 1 when TIMER A starts

run — This sets the runmode. When 0, TIMER A will operate in one-shot mode, i.e it will run for one countdown and then stop When the bit is 1, TIMER A will operate in continuous mode

load — You can only write a 1 to this bit When you do, TIMER A is re-loaded with it's preset value regardless of it's current state.

inm — If this bit is zero, TIMER A will count clock pulses. When set to 1 the timer will count (positive) pulses applied to the CNT pin. Thus you may use TIMER A as a timer (system clock) or a counter (CNT).

spm — This bit sets the Serial Port mode for input (bit=0) or output (bit=1). We will look at the serial system later.

tod — Because the 64 is sold all over the world and some countries use mains supplies of differing frequencies, the Time Of Day clock is able to operate at different speeds. This bit sets the TOD source clock to either 50Hz (bit=1) or 60Hz (bit=0) In this country, we set this bit to 1. We will look at the tod clock shortly.

Neither TIMER A or TIMER B of CIA 2 are used by the system so they are available for our use. It's probably best to leave CIA 1 alone except perhaps to turn off the system IRQ timer Since Port B appears at the user port, you have both PB6 and PB7 lines and the CNT lines from both CIA s

Program 7 shows a simple timer-based, interrupt driven program which links timer's A and B together to produce an interrupt approx once every second to change (increment actually) the screen border colour. Not very exciting stuff, but it does show you one way to use the timers

Control Register B

bit→	7	6	5	4	3	2	[illegible]	[illegible]
	alarm	inm1	inm2	load	run	out	pbon	sta[illegible]

Bits 0-4 are the same to those in CRA except that they control TIMER B [illegible]ather than TIMER A and TIMER B out-[illegible]ut appears on PB7

[illegible]m1/2 TIMER B may use one of four possible inputs. These are as follows:

inm 1	inm2	
[illegible]	0	TIMER B counts system clock pulses.
[illegible]	1	TIMER B counts positive CNT pulses.
[illegible]	0	TIMER B counts TIMER A underflow events.
[illegible]	1	TIMER B counts TIMER A underflow events only when CNT is held high.

alarm — When this bit is zero, writing to the TOD registers sets the TOD clock whereas when this bit is 1, a write to the TOD registers sets the alarm.

Program 7. Simple CIA Timer interrupt

```
[illegible]LE CIA [illegible]
[illegible] $CB00
[illegible]QA $DD00
[illegible]
JMP START
JMP STOP
;
 START SEI
LDA #<INTR
LDX #>INTR
STA $0318
STX $0319
LDA #<1000
LDX #>1000
STA CIA+4
STA CIA+6
STX CIA+5
STX CIA+7
LDAQ #$0
LDX #$41
STA CIA+14
STX CIA+15
LDA #$82
STA CIA+13
CLI
RTS
.
.STQP
SEI
LDA #$1F
STA CIA+13
LDA #$47
LDX #$FE
STA $0318
STX $0319
LDA #$00
STA CIA+14
STA CIA+15
CLI
RTS
:
.INTR PHA
TXA
PHA
TYA
PHA
LDA CIA+13
AND #$02
BNE 3
JMP $FE4C
INC $D020
JMP $FEBC
```

```
READY
```

Program 7 Operation

You can start this routine by typing SYS 51200. I've put it here so as not to interfere with the Parallel Printer Driver if you have it in memory You can always relocate it if you wish.

Both timers are set to a pre-scaled value of 1000 Since TIMER B will be counting TIMER A underflow events and TIMER A counts system clock pulses (1 million per second), the resultant delay should be around 1 second.

CRA is set to $01 (start TIMER A) and CRB is set to $41 (start TIMER B and set it to count TIMER A underflow events) In order to enable the TIMER B interrupt, we write $B2 to the ICR

When TIMER B reaches zero, the NMI interrupt service routine takes control As before, we check that the interrupt has been caused by our timers Once we have incremented location $D020 (the VIC II border colour register) we return via the exit interrupt routine at $FEBC Use either RUNSTOP/RESTORE or SYS 51203 to stop it.

Time of Day Clock

The TOD Clock is split over 4 consecutive registers starting at base $+$04 The registers all read out in BCD format which simplifies their use The organisation is as follows.

base+$04	Tenths of seconds
base+$05	Seconds
base+$06	Minutes
base+$07	Hours

For the clock to count accurately in this country, you must set the tod bit in Control Register A (CRA) to one This sets the clock rate for 50Hz which is the mains frequency in this country Bit 7 of the Hours register is the AM/PM indicator (1=PM) and may thus be tested easily with the BIT instruction.

The following procedure should be adhered to when setting or reading the clock When setting the clock, TOD is automatically stopped when a write to the hours register takes place and will restart again after a write to the tenths register So, if you write hours-minutes-seconds-tenths (in that order) then the clock will always start at the desired time.

When reading the clock, you should read in the same order. Although the timer keeps counting during the read operation, the values are temporarily latched when the hours register is read.

The alarm may be set by first setting the alarm bit of Control Register B (CRB) to 1, then writing the alarm values to the TOD registers Remember to reset the bit to 0 afterwards

Program B uses the TOD clock in CIA 2 to display a clock on the screen It starts at 13:00.00 and the alarm is set to 13:00.01 The program simply loops around displaying the clock values. When the alarm occurs (via an NMI interrupt) the program will stop. You could, for example, use the system 1/60th second interrupt to do the display work thus allowing the machine to get on with some other work.

Notice here that I have not set the TOD frequency bit (CRA-bit 7) as I described earlier For this demonstration it doesn't matter too much but you should make sure you set it correctly when measuring longer periods of time.

```
 TIME OF DAY CLOCK EXAMPLE
;
ORG $C400
;
.CIA EQA $DD00
;
 START JSR INIT
JSR DISPLAY
JSR STOP
RTS
;
 INIT SEI
LDX #<INTR
LDY #>INTR
STX $0318
STY $0319
LDX #0
STX CIA+15
LDX #3
.SLO LDA INI1CLOCK,X
STA CIA+8,X
DEX
BPL SLO
LDX #$80
STX CIA+15
LDX #3
 SL1 LDA INITALARM,X
  A CIA+8,
DEX
BPL SL1
LDA #$B4
STA CIA+13
LDA #0
STA ALARM
CLI
RTS
;
.INITCLOCK BYT 0,
.INITALARM BYT 0,
;
.STOP SEI
LDA #$1F
STA CIA+13
LDA #$47
STA $0318
LDA #$FE
STA #0319
CLI
RTS
;
 .ALARM BYT 0
;
 INTR PHA
TXA
PHA
TYA
PHA
LDA CIA+13
AND #$04
BNE 3
JMP $FE4C
LDA #$80
STA ALARM
JMP $FEBC
;
 CLOCKDATA BYT 0 0,0,0
;
 DISPLAY
 DL1 LDX #3
 DL2 LDA CIA+8 X
STA CLOCKDATA,X
DEX
BPL DL2
LDX #1
LDY #1
CLC
JSR $FFF0
LDX #3
.DL3 LDA CLOCKDATA,X
AND #$7F
JSR DATAOUT
DEX
BPL DL3
LDA #'A
BIT CLOCKDATA+3
BPt 2
LDA # P
JSR $FFD2
BIT ALARM
```

```
        [illegible]
        RTS
        ;
        .DATAOUT P[illegible]
        LSR A
        LSR A
        LSR A
        LSR A
        ORA #$30
        JSR $FFD2
        PLA
      0 AND #$0F
        ORA #$30
      0 JSR $FFD2
      0 LDA #' '
        JMP $FFD2
      0 ;
```

Program 8 Operation

The program will initialise the CIA registers and will then run in a loop which reads the TOD registers and displays the result on the screen. This loop will operate until the flag byte ALARM is set to $80 by the NMI interrupt service routine which will take control when the TOD alarm operates

The byte strings INITCLOCK and INITALARM set the initial TOD conditions They are "back-to-front", i e 10th second byte first Remember that the 7th bit of the hours register is the AM/PM flag Here both the time and alarm hours byte have bit 7 set to 1

As described above, each register reads out in BCD format so all we have to do with each byte in order to display it's value is to logically OR each nybble with $30 hex thus turning it into an ASCII character We read the clock registers into the CLOCKDATA byte string then the routine DATAOUT performs the conversion operation on the byte passed to it in the accumulator.

When the alarm interrupt occurs, the ALARM byte is set to $80 When this happens, control will pass from the DISPLAY subroutine and back to BASIC via the STOP subroutine which will disable the CIA

The Serial Port

This is a buffered, 8-bit register system The serial control bit in CRA selects input or output mode. I should point out that this isn't an RS232/432 etc serial format If you want this then you should use the kernal's RS232 system calls.

In input mode, data applied to the SP pin is shifted into the shift register on the positive edge of the signal applied to the CNT pin After 8 CNT clocks, the data is passed into the serial data register and a Serial event signalled in the ICR, bit 3. If enabled, this will cause an interrupt

In output mode, data is shifted out on the SP pin at half the underflow rate of TIMER A which is used as the shift-rate clock The shift register will begin operating as soon as the data byte is written to the Serial Data Register, assuming TIMER A is running and is in continuous mode Once all 8 bits have been shifted out, a Serial event is signalled to indicate more data may be sent If data has already been written to the SDR (the processor stays 1 byte ahead of the CIA) the data flow will be continuous

The maximum data shift rate (baud rate) possible is the system clock rate divided by 4. In our case this is about 250,000 bits per second (more on a C128 in 128 mode). This maximum rate may be affected by what you have connected to the SP line and how fast your receiving device can accept the data

The most interesting aspect of the Serial system is the ability to connect several CIA's together on a common serial communications bus Both the SP and CNT pins are able to be connected in such a fashion.

Let's assume that you have several computers (64's or 128's) whose serial CIA outputs are connected together. In order to regulate the flow of data, one CIA must act as a "master" device. You could for example have only one disk drive connected to the master computer and be able to exchange programs between this and the other machines The master thus becomes a "File-Server"; accepting requests to load or save programs from 'client" machines. What you end up with is a computer network.

There are a number of ways to approach the problem of writing a network operating system One method is to have the File Server to send enquiry messages to each client in turn to ask if it requires attention If the client requires a network service, it may now tell the fileserver what it needs, e g. load a program, save a program, etc

This is fine assuming that the clients do not require continuous attention from the server since it is possible for one machine to grab the network of an extended period of time, locking out the other clients. It also means that if the server has a number of machines to cater for, it could be some time between each enquiry to a particular machine.

Another method is for a client machine to send a message to the server when it requires attention. This leads to a more efficient "as needed" service for the clients The disadvantage is that there is the possibility that two or more machines may try to communicate with the server at the same time Some way must be found for a client to detect that someone else is in communication with the server, and back off until the line is clear. One way to do this would be to set the otherwise unused TIMER B to check if the CNT pin is active. We can program CRB, and therefore TIMER B, to count CNT positive pulses If we set up TIMER B just before we attempt to send a message, we can check to within a few microseconds of transmission if anything is happening on the bus.

How would each machine know that a message being sent is meant for it? Well, each client machine would have an ID number (station number) and every message sent over the bus would be coded with the target machine's ID number It may also have the ID of the transmitting machine to identify the party responsible This means that there are some operations which the fileserver would not be involved with such as sending a text message from one computer to another

When using the serial system, we are concerned mainly with the following registers,

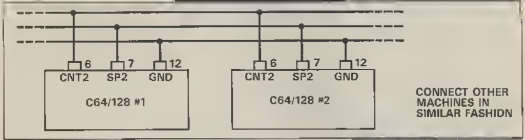
Serial Data Register	base+$0C
Timer A register pair	base+$04
	base+$05
Control Register A	base+$0E
Interrupt Control Reg	base+$0D

Bit 6 of CRA controls the serial port mode, 1=output, 0=input Bit 3 of the ICR is the serial event flag/enable. The TIMER A pair should be programmed with the value required for the shift rate!

Although the data flow rates would be matched as closely as possible, they need not be exactly the same since it is the transmitting machine that outputs the signal that times the transfer

Serial Port Programming

Unfortunately, unless you have three or more machines available, there isn't really much point in setting up a network What I will do here is to show some basic programming concepts for the serial system The technique used to program the serial port is not unlike that used for the parallel port described earlier



For the transmit phase, the port is set for output, you write a data byte to the data register and wait for a flag in the ICR to indicate data may be sent again For the receive phase, the port is set for input, you wait for a flag in the ICR to indicate data has arrived and then you read the byte from the data register. As with the parallel system, the serial system may use interrupts to perform the data transfers to and from the data register. The major difference is the use of TIMER A to clock the transfer.

Take a look at Fig 9 Programs 9 and 10 are similar to programs 1 and 2 in that they are simple BASIC data transfer programs, but using the serial port instead I have used a moderate transfer rate and you can always try to speed it up by reducing the values in lines 20 and 30 of program 1, but remember that, unlike the parallel system, the serial system has no control lines and it is therefore possible to lose data if your receiver cannot process the event quickly enough. To be honest, this is best handled by machine code, but these programs may help you see how the system works.

Fig 9. Serial Transfer Programs

```
Program 9
10   CIA=56576
20   POKE CIA+4,255
30   POKE CIA+5,0
40   POKE CIA+14.65
50   INPUT R$
60   R$=R$+CHR$ (13)
70   FDR I=1 TO LEN (R$)
80   POKE CIA+12,ASC (MID$
     (R$,I,1))
90   IF (PEEK (CIA+13) AND 8)=0
     THEN 90
100  NEXT I
110  GDTO 50
```

These programs show the basic method of using the serial port.

```
Program 10
10   CIA=56576
20   POKE CIA+14 0
30   IF (PEEK (CIA+13) AND 8)=
     OTHEN30
40   PRINT CHR$ (PEEK (CIA+12));
50   GOTO 30
```

Programming the Serial System Using Interrupts

If you did go on to write some sort of Network Filing System (NFS) then you would probably need the serial port to be interrupt driven. On the 64, you have to access to serial ports on both CIA 1 and CIA 2 Using CIA 2 is probably wisest since it generates NMI's rather than IRQ s.

Program 11 allows input from the keyboard and then sends it out via the serial port under interrupt. Program 12 receives data from the serial bus under interrupt and isn't all that different from the parallel version (program 4) You could try converting it to operate the parallel port instead

Both programs live at $C000 and should be started by SYS 49152

```
[illegible]
[illegible]
[illegible]  ND [illegible] R INTER [illegible]
[illegible]
[illegible]  G $C [illegible]
[illegible]
[illegible]  ,CIA EQA $DD00
[illegible]  ;
[illegible]  ;
[illegible]  ,START JSR INIT
[illegible]0 ,GETIN
[illegible]20 JSR INPUT
130  LDA #$80
140  STA SFLAG
[illegible]  [illegible] Y #0
[illegible]  LDA BUFFER,Y
[illegible]  CMP # *
[illegible]  BEQ  EXIT1
[illegible]  STA CIA+12
[illegible]   WAIT BIT SFLAG
[illegible]  BMI WAIT
220  JMP GETIN
230   EXIT1 JMP STOP
240  ,
250   INIT SEI
260  LDA #<INTR
270  LDX #>INTR
280  STA $0318
290  STX $0319
300  LDA #$88
310  STA CIA+13
320  LDA #$FF
330  LDX #$00
340  STA CIA+4
350  STX CIA+5
360  LDA #$41
370  STA CIA+14
380  LDA #0
390  STA SFLAG
400  CLI
410  RTS
420  ,
430   INPUT LDY #0
440  IN1 JSR $FFCF
```

```
        INY
        CMP
        BNE I
        JSR $
        LDY #I
        STY IND
        RTS

       SEI
      #$47
     X #$FE
     A $0318
    STX $0319
    LDA #$7F
    STA CIA+13
    CLI
    RTS
    ;
    .INTR PHA
    TXA
    PHA
    TYA
    PHA
    LDA CIA+13
    AND #$08
    BNE 3
    JMP $FF4C
    BIT SFLAG
    BMI 3
    JMP $FEBC
    INC $D020
    LDY INDEX
    LDA BUFFER,Y
    STA CIA+12
    INY
    STY INDEX
    CMP #13
    BNE INTRI
    LDA #0
    STA SFLAG
    .INTRI JMP $FEBC
    ;
    .BUFFER RES 80
    .SFLAG BYT 0
    .INDEX BYT 0
    ;
    Y.


    ;
    ; CIA SERIAL
    ; RECEIVE UNDER INTERRU
    ;
    ORG $C000
    ;
    .CIA EQA $DD00
    ;
    ;
    .START JSR INIT
    .WAIT JSR $FFE1
```

```
    STA
    JMP W
    .EXIT1 J
    ;
    .INIT SEI
    LDA #0
    STA SFLAG
    STA INDEX
    STA CIA+14
    LDA #<INTR
    LDX #>INTR
    STA $0318
    STX $0319
    LDA #$88
    STA CIA+13
    CLI
    RTS
    ;
    .STOP SEI
    LDA #$47
    LDX #$FF
    STA $0318
    STX $0319
    LDA #$7F
    STA CIA+13
    CLI
    RTS
    ;
    .OUTPUT LDY #0
    .OUTI LDA BUFFER,Y
    JSR $FFD2
    CMP #13
    BEQ OUTEND
    INY
    BNE OUTI
    .OUTEND RTS
    ;
    .INTR PHA
    TXA
    PHA
    TYA
    PHA
    LDA CIA+13
    AND #$08
    BNE 3
    JMP $FE4C
    LDA CIA+I
    LDY INDEX
    STA BUFFER,Y
    INY
    STY INDEX
    CMP #13
    BNE INTRI
    LDA #$80
    STA SFLAG
    .INTRI JMP $FEBC
```

```
    FLAG
    INDEX
```

Program 11 Operation

The program initialises in the routine INIT. The NMI vector is altered and the CIA2 serial interrupt enabled. The SFLAG variable is used to signal to the NMI routine that data is ready to go out and we initially set it to zero. We also set and start TIMER A and set the serial port for output.

After data is received from the keyboard, the data index pointer is set to 1. SFLAG is set to $80 (data ready) and we store the first data byte to the serial data register in order to start up the system. When the last character has been sent (always 13), SFLAG is reset to zero and the process begins again with keyboard input.

You may stop the program by entering an asterisk as the first character on a line.

Program 12 Operation

Like program 11, this one initialises the serial system except that we set it for input. The program waits for a line of data to be received via the NMI interrupt then prints it out. SFLAG indicates the current state of affairs, SFLAG = 0, no data; SFLAG = 1, data received and ready.

Of course, you don't have to keep looping around like I have done here. You can periodically poll the byte to see if data has arrived and take any necessary action.

I hope you will be able to see the possibilities afforded by the serial system. The examples will, I hope, help you understand how it works. Writing network system software is a challenge for most of us and I wish you luck if you do try it. Your biggest problem may be getting hold of enough machines to test it out!

That just about wraps up the 6526 CIA. The examples I've given you are perhaps not the most exciting in some cases but I believe they do show how to program the device. Next time, we will take a look at the Plus 4's facilities and there may be a surprise or two in sore for some of you.

Codemaster

Andy Partridge looks at a comprehensive machine code utility from a new development house

The North of England seems to produce more software companies than anywhere else on this fair island of ours. The latest is a company called 'Diamond Bytes', specialising in utility programs of good quality at a reasonable price. Their first release is a machine code utility called 'Codemaster'. This program is quite a comprehensive Debugger, Analyser, Tracer, Monitor and Assembler package.

I must admit to never really using a Debugger before this. I always took a printout of my code and skipped through it by hand! Codemaster is an excellent utility that does this for you, and I'll be using it instead of 5 feet of printer paper from now on!

To use Codemaster, you first get your machine code program into memory, either by loading the file in straight or assembling, whatever you need to do it doesn't matter. You could even reset a commercial product and have a peek at what's going on! Next you load in codemaster and run it. If codemaster is somewhere you don't want it to be in memory, a command will make it move itself to somewhere else to avoid memory clashes. Once you've loaded and run Codemaster you will get 10 lines of disassembled code, the address Codemaster is at, the contents of the X, Y and A registers and the Status register shown and a command line. All the 6510's flags and registers can be changed from within the program, a CLC command will clear the carry, A SEI will set the interrupt flag etc. and the A, X and Y registers can be altered with any LDA/LDX such as LDA $AOOO,X or LDA ($34),Y they all work! In fact, you can enter any 6510 instruction from within Codemaster except the branching commands. 'Windows' can also be opened to show ASCII in memory.

If you want to 'Run' the program that is in memory, you must first go to the start of it with a JMP command. The 10 line display will then show the first ten lines of your program. You can single-step through the program with the RETURN key, or by pressing F1 which will run until an RTS is met. This is useful for by-passing a subroutine that you know works, or to skip a Kernal routine (Which you hope works!!) As Codemaster passes an instruction, it carries it out. All the registers are changed to show you what state the processor is in at that time, and any branches, JSR's or JMP's make Codemaster jump to that section of memory and display it.

If you want to stop your code at a certain address, TRAPS can be set to halt the progress of Codemaster through your program. Alternatively, Run/Stop can be used to halt the flow of a program (if you are quick enough to stop it where you want it!)

Another great feature of this program is that it lets you assemble code into memory, entering a PTC $XXXX instruction points the 10 line assembling at the specified area of code, and then all instructions entered will be assembled into memory. Illegal expressions won't assemble, and you can jump back and step through it in the same way as above.

If the screen gets corrupted by the program you are running, pressing F5 will get it back. F7 performs a warm start and F1, as I said above, runs a subroutine.

The manual comprises of an introduction, an overview of the 6510 processor and then a series of tutorials to teach you how to use the program. The tutorials teach well, with the exception of a lack of explanations for the F-Keys in an easy list. . They are dotted through the text. The whole instruction manual is quite easy to read, with three examples for you to look through on the disk to get you started.

And to end with, a list of features you may be interested in as a serious user:

+ Memory Hex dumper
+ Machine code Trace Utility
+ Single step or Walk mode
+ Sequence run mode
+ Direct machine code mode
+ Code editing mode
+ External patching mode
+ External program access
+ Independence from Kernal Screen, Keyboard and Interrupt routines

To be fair I have not given this program it's due justice. A lot more could be said of it's capabilities. It is a long time since I have had such a good utility in my hands and for it's price it must be considered a must for all serious machine code programmers.

AT A GLANCE
Title: .Codemaster
Suppliers: Diamond Bytes, 7 Graham Avenue, Brinsworth, Rotherham, SG0 5LA.
Price: £15.95 (Disk only)

Personal Organiser Page Printer

Produce your own personal pages with the same format as most personal organisers especially for the C128

By Paul Traynor

Personal Organiser Page Printer (POPP) is a program which enables the user to produce pages which match the format of common organisers You have the ability to give your pages a main title and footer along with up to 3 separately titled columns. Page details can be stored on disk for later retrieval. A Commodore C128, 80 column monitor and printer are essential

Operation

Use of POPP is very simple with just 2 menus, the first has 5 options;
1. Create Page
2 Print Page
3. View Page
4 Disk Functions
5 Exit Program

After selection of the Create function you will be shown a representation of the current page in memory (if any), you are then prompted for the title information starting with the main title, then the number of columns (1-2) and then their titles and finally the footer

The Print function will put the completed page on paper The View function allows you to view a scaled representation, scaled so that it will fit onto one screen. The function will cut down unnecessary paper usage.

Selecting Disk Functions will take you to the disk functions menu. Exit Program, you are then asked if you are sure for safety

The Disk Functions menu has 4 options,
1. Device Number and Disk Directory
2. Save Page
3 Load Page
4 Exit disk Functions

Option 1 allows you to choose the device number (default is 8) and also displays the directory of sequential files that begin with POPP. The Save function will store the current page details to disk. If a file exists with the same name then it will be replaced.

The load function will load a page into memory When using either load or save you do not have to enter the filename prefix (POPP). Exit will take you back to main menu

Printouts

POPP should work with any Commodore compatible printer. The final printout will be a lined page with your titles and footers (Printouts will be in near letter quality if your printer supports it). The page just needs to be cut out and the 6 holes need to be punched in the positions which are marked. On the disk are some example pages which show what can be done with Personal Organiser Page Printer

C128 Convertor and Maths Aid

We give you a couple more small but useful C128 programs and put practical theory to the test

By Paul Traynor

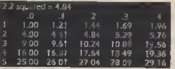
Convertor and Maths Aid has, as the name implies, two distinct parts. The first is a full functions metric and imperial converter, part two is an electronic Log and Data book Functions include support for disk drive and CBM printer.

Converter

The six converter options can be selected from the main menu, these are length, area, volume, mass, velocity and temperature. These are all standard metric and imperial conversions except temperature which is a three way conversion, ie Celsius, Fahrenheit and Kelvin. After picking your selection from these pages you are given a submenu showing all the possible calculations etc inches to millimetres or cubic metres to cubic feet After selecting your particular conversion you are prompted for an input value and after pressing return you will be given the solution At this point you can either do another conversion of the same type or return to the main menu

Maths Aid

After selection of this part of the program from the main menu, you are presented with the following options. trigonometry, functions; logarithms, squares, roots and reciprocals; and degrees and radians Each one will then lead to its own menu and these will show the actual calculations available. For example Natural Logarithm or Inverse Tangent. If you are working with the trigonometry functions it will be necessary, when prompted, to enter whether you are using degrees or radians for measuring your angles The main difference between this part of the program and the Converter part is that after giving your answer, it goes on to show it in tabular form like an extract from a Log Book This will be up to 25 calculations down the left hand column will be integer parts of inputed values, along the top row will be the decimal parts of inputted values An example of the tabular form of answer is shown below



```
2.2 squared = 4.84
       .0      .1      .2      .3      .4
1    1.00    1.21    1.44    1.69    1.96
2    4.00    4.41    4.84    5.29    5.76
3    9.00    9.61   10.24   10.89   11.56
4   16.00   16.81   17.64   18.49   19.36
5   25.00   26.01   27.04   28.09   29.16
```

To use the table you line up the row which begins with the integer and the column headed by the fraction, the point where they cross will give you your answer If you are calculating an inverse cosine or similar then the actual value that you input will be in the main body of the table while the answer will be formed from the first row and the first column

Disk and Printer Functions

You can have a hardcopy of the Conversions and/or the Maths Aid calculations that you have carried out by selecting these options from the menu You can also have a copy of your conversions or calculations, to disk, as a sequential file which can then be read into your word processor. For saving to disk you will have to enter a device number and a filename.

You will find the programs associated with the above on the disk Make sure you are in C128 mode before trying to use them

1581 Direct Access

Following on from the 1581 toolkit review, Paul Traynor puts some practical theory to the test

1581 Direct Access

The Direct Access commands provide a means to read, write and alter the data on your disks by track and sector. Using Direct Access commands gives greater flexibility when programming but generates more work for the programmer, who has to perform file management unlike when using program, relative or sequential files when data is already DOS organised. Uses for Direct Access commands stretch from simple locking/unlocking files to creating a full blown data management program to suit your needs.

General Rules

Firstly it is wise to use a blank disk to avoid corrupting data by mistake. It is a good idea to have a sector editor close at hand, these can be extremely useful in carrying out simple tasks. The one supplied on the 1581 Demo/Utilities disk is adequate. It is also a good idea to check the error channel after rear or write operations.

Direct Access Commands

There are several different commands for block-read and block-write, these are split into three distinct types, below these are listed in the same order as in your 1581 user's guide:

1. User commands

READING

Firstly we have u1 or ua these commands can be executed in any of the following formats, they allow the user to read a complete block. u1 can also be replaced by ua

```
PRINT#15,"U1"; CHANNEL #, DRIVE
#;  TRACK #;SECTOR #
PRINT#15,"U1  CHANNEL #, DRIVE
#,    TRACK #,SECTOR #"
PRINT#15,"U1," CHANNEL #  DRIVE
#, TRACK #;SECTOR #
```

The program shows an example of the block read command. You can read any track and sector of your disk and it will be printed on screen. It is assumed throughout that the 1581 is set at device number 9, if this is not so, the open statements in lines 20 and 30 should be altered.

```
10 INPUT "ENTER TRACK TO READ":
TR:INPUT "ENTER TO READ"; SE
20 OPEN 15,9,15
30 OPEN 5,9,5, "#0"
40 PRINT # 15, "U1", 5, 0, TR; SE
50 FOR I=1 TO 256
60 GET#5, A$:IF A$=""THEN
A$=CHR$(0)
70 POKE I + 7936, ASC
(A$):PRINTA$,
80 NEXT
90 CLOSE5.CLOSE15
```

WRITING

Similarly we have u2 or ub for writing whole blocks. Again there is a host of different formats to choose from. u2 can also be replaced by ub.

```
PRINT#15, "U2"; CHANNEL #,DRIVE
#,  TRACK #;SECTOR #
PRINT#15, "U2:CHANNEL #,DRIVE
#,     TRACK #,SECTOR #"
PRINT#15,"U2" CHANNEL #,DRIVE
#,    TRACK #;SECTOR #
```

The program shows an example of the block write command. You can write data to any track and sector of your disk up to a maximum of 256 characters per block

```
10 INPUT"ENTER DATE TO BE
WRITTEN"; NA$
20 INPUT "ENTER TRACK"; TR
30 INPUT "ENTER SECTOR"; SE
40 OPEN 15, 9, 15
50 OPEN 4, 9, 4, "#"
60 PRINT #15, "B-P";4;0
70 PRINT#4, NA$
80 PRINT#15,"U2",4;0;TR;SE
90 CLOSE4.CLOSE15
```

All of the U commands will check for errors in the designated track and sector and will report any errors in user's input

2, No error checking

Reading

Secondly we have b-R, this command will allow the user to read a complete block but it does not perform any error checking. Below are the formats for this command which can be used in the first block read program by replacing line 40.

```
PRINT#15, "B-R";CHANNEL #;DRIVE
#,  TRACK #;SECTOR #
PRINT#15,"B-";CHR$(210)CHANNEL
#,DRIVE #,TRACK #,SECTOR #
PRINT#15, "B-R " CHANNEL #,
DRIVE #,TRACK #;SECTOR #
NOTE. R IS SHTED
```

Writing

Similarly we have b-W for writing whole blocks. Once again the formats below can be used in the first block write program by replacing line 80

```
PRINT#15, "B-W"; CHANNEL #,DRIVE
#;TRACK #,SECTOR #
PRINT#15, "B-",CHR$(215),CHANNEL
#;DRIVE #,TRACK #;SECTOR #
PRINT#15, "B-W: "CHANNEL #;DRIVE
#,TRACK #,SECTOR #
NOTE: W IS SHIFTED
```

3, Original Commands

Reading

Thirdly we have the so-called original block read command, with this command it is not possible to read a whole block because the first byte is read as a marker to determine how much of the block is to be read. Two formats for the command are shown along with an example program. In the example the variable st is used to indicate the end of the data when it changes from 0 to 64

```
PRINT#15,"B-R";CHANNEL
#;DRIVE #;    TRACK #; SECTOR
#
PRINT#15, "B-R:" CHANNEL
#;DRIVE #; TRACK #; SECTOR #
10 INPUT "TRACK TO READ"; TR
20 INPUT "SECTOR TO READ"; SE
30 OPEN 15,9,15
40 OPEN 5,9,5, "#0"
50 PRINT#15, "B-R";5;0;TR;SE
60 FOR I=1 TO 256
```

```
70 GET#5,A$:IF A$=""THEN
A$=CHR$(0)
80 POKE I+7936,ASC(A$)
90 printa$;:if st=64 then 110
100 next
110 close5:close15
```

Writing

We have a corresponding original block write command where the first byte is set as a marker to indicate the number of bytes recorded in the block

```
PRINT#15,"B-W'; CHANNEL #,
DRIVE #, TRACK #,SECTOR #
PRINT#15, "B-W·' CHANNEL
#,DRIVE #, TRACK #,SECTOR #
10 INPUT "ENTER DATE TO
WRITTEN";NA$
20 INPUT 'ENTER TRACK",TR
30 INPUT "ENTER SECTOR";SE
40 OPEN 15,9,15
50 OPEN 4,9,4,"#'
60 PRINT#4,NA$
70 PRINT#15,"B-W",4,0TR,SE
B0 CLOSE4
90 CLOSE 15
```

It is important to note that, although it is possible, it is not at all advisable to mix block commands and original block commands in the same program

Buffer Pointer

Setting the buffer pointer determines where any reading or writing will commence in the block This is used for accessing individual bytes and can also allow the user to organise the blocks into fields similar to relative files. Three allowable formats are shown

```
PRINT#15,"B-P";CHANNEL #,BYTE
PRINT#15,"B-P:CHANNEL #,BYTE'
PRINT#15,"B-P:"CHANNEL #,BYTE
```

The example program allows the user to change the id ol any disk on the 1581. The disk id number is stored in three places on the disk these are all catered for in our example which shows the buffer pointer command at work.

```
10 INPUT "ENTER NEW ID· ';ID$
20 OPEN 15,9,15
30 OPEN 5,9,5, "#'
40 PRINT#15, "U1";5;0,40;00
50 PRINT#15, 'B-P";5;22
60 PRINT#5,ID$,
70 PRINT#15, "U2";;5;0;40;00
80 PRINT#15,"U1 ',5,0,40;01
90 PRINT#15,"B-P";5,04
100 PRINT#5,ID$;
110 PRINT#15,"U2";5;0;40;01
120 PRINT#15, "U1 ";5,0,40,02
130 PRINT#15, "B-P",5,04;
140 PRINT#5,ID$;
150 PRINT#15, 'U2",5,0,40;02
160 CLOSE 5
170 CLOSE 15
```

Allocating Blocks

This allows the user to mark the blocks which he has used so that they will be written over by other disk accessing such as saving program files, however, these will be cancelled by a validate or collect command. Two command formats are shown

```
PRINT#15, "B-A"; DRIVE #;TRACK #
,SECTOR #
PRINT#15,'B-A... 'DRIVE #,TRACK #;
SECTOR #
```

The user guide provides a good example of the BlockAllocate command, which will select the first available block and then tell you which one it has chosen. Our example allows you to input your track and sector and will then report any errors.

```
10 INPUT "ENTER TRACK TO ALLO-
CATE: ";TR
20 INPUT "ENTER SECTOR TO ALLO-
CATE. ",SE
30 OPEN 15,9,15
40 OPEN5,9,5"#"
50 PRINT#15, "B-A",0,TR;SE
60 PRINT EN,EM$,ET,ES
70 CLOSE5  CLOSE 15
```

Freeing Blocks

This command allows a previously allocated block to be freed It works by updating the block availability map. As with block Allocate two command formats are shown.

```
PRINT#15,"B-F";DRIVE #;TRACK #,
SECTOR #
PRINT#15, 'B-F...' DRIVE #;TRACK #,
SECTOR #
```

Once again our example is a simple one which allows input of track and sector to be freed

```
10 INPUT "ENTER TRACK TO FREE.
';TR
20 INPUT "ENTER SECTOR TO FREE:
";SE
30 OPEN 15,9,15
40 OPEN 5,9,5,"#"
50 PRINT#15, "B·F";0;TR,SE
60 PRINT EN,EM$,ET;ES
70 CLOSE5' CLOSE15
```

Partitions

The one great advantage of the 15B1 drive over its predecessors is its ability to create partitions on the disk. Uses for partitions include reserving space for your own data manipulation programs, and probably the most popular sub-directories

We have already discussed the Allocating Block command and how it will be overwritten by a validate or collect command Partitions provide a work area which will not be overwritten by such an Instruction The best way to produce partitions is to use the Partition Aid utility supplied, but below are the commands for addition in your own programs

```
PRINT#15,"/0 PARTITION
NAME, '+CHR$ (TRACK#) + CHR$
(SECTOR#) + CHR$ (X) + CHR $(Y) +
",C'
WHERE
Y=INT (N/256)
X=N·Y*256
N=NUMBER OF BLOCKS REOUIRED
IN PARTITION
```

The following command will select a partition from the root directory

```
PRINT#15,"/0:PARTITION NAME'
```

You can use the partition as it stands, with no sub-directory, but you will have to use the block read and block write commands which do not check for errors as mentioned before.

After creating and selecting a partition you can format with the following command

```
PRINT#15,"N0.PARTITION NAME,ID
```

If you wish to create a sub-directory for your partition it must comply with the following rules:

1. The partition must be at least 120 blocks in size.
2. The first sector must be 0.
3. The total number of sectors must be a multiple of 40.
4. The partition area must not contain track 40.

Below is a simple program which allows the user to produce any size of sub-directory with the start at track 1.

```
10 INPUT 'NUMBER OF BLOCKS RE-
OUIRED IN PARTITION",N
20 OPENS 5,9,5
30 OPEN 15,9,15
40 Y=INT (N/256)  X=N·Y*256
50 PRINT# 15,"/0 :PARTITION
NAME, "+ CHR$ (1)  + CHR$ (0)+
CHR$ (X) + CHR$ (Y) +",C"
60 PRINT#15,  '/0:PARTITION
NAME"
70 PRINT#15, "N0:PARTITION
NAME,ID"
```

Finally

On the disk are B very short demonstration programs covering the topics mentioned here. Make sure you put your computer into 128 mode before attempting to look at them.

Muncher

Just when you thought you had seen the last of the Pacmen they reappear more hungry than ever

By David Bryson

The scientists at Washington are a clever bunch. After genetically engineering an animal to get rid of the current drugs pushing problems, you have taken form as a fat yellow beast nicknamed the Muncher. Your mission Jim, is to eat all these horrible drugs lying around the Washington depots and elminate the crisis for years to come. However in these depots remains the ghosts of earlier drug takers, who certainly don't like you hanging around. But not to worry.. the scientists devised a plan of sending fake dealers into these depots to scatter plasma capsules all over the building. (You see the drug pushers didn't notice the difference).

Playing Muncher

To play Muncher, you move your man around the maze gobbling all those nasty drugs, and avoiding those nasty nasties. However, when you eat a plasma pill, you can get your own back on these horrible ghosts by gobbling them up. But unfortunately the plasma energy in the pills is only sufficient to last up to 5 seconds.

Muncher is played with the joystick in PORT 2 using the normal movements (ie left to move left, right to move right etc.), but the FIRE button is NOT used. If you can't hack the pace, you can PAUSE the game with any key (except COMMODORE and F7 which changes the SPEED of the game, and RUN/STOP which QUITS the game). To continue, press any key. In total there are 15 depots of action to complete, with a BONUS-BUILDING at the end. At the start you have 4 men at your disposal. Can you munch your way to the last level without being caught four times? (By the way, this game has been protected against disabling collision, etc. with cartridges, so don't try to cheat!)

The high score table

Well, if you don't manage to save the world, don't worry, your name's still going up in lights. To enter your name on the Hall of Fame, you move the joystick up or down to select a letter, and then press fire if it's the one you want. To finish quickly, you can press right and this will act as a Return Key.

This is the first game I have made (at 14) and so I hope that you enjoy it. May Muncher live for thousands of years to come! Right, now stop reading, and get playing the game! (Hellos go to SCUM U K and VIDEO1-FM, from THE NAMELESS ONE of REVOLUTION).

MYTH

Gordon Hamlett examines the latest from System 3 software.

Every civilisation has its own supply of myths and legends. The Gods in their Pantheon, Olympus to the Greeks, Valhalla for the Norsemen and so on. These deities controlled the fate of every mortal on Earth. Heroes performed great feats or bravery against evil demons and malevolent spirits. But we don't believe in that sort of rubbish any more do we? Life today is more concerned with mundane matters such as whether Manchester City will stay in the First Division or how to cope with the ever increasing number of junk fax messages that plague the lives of those rich enough to afford a fax machine in the first place.

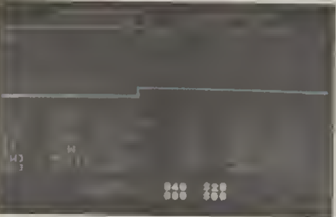
One boy believed though. On the face of it, a perfectly normal youth, popular with his friends at school but the bane of his history teacher's life for his abnormal obsession with classical mythology. As it happened, the Titans (for they do exist) were looking for such a person. Someone who believed in them totally and utterly. The evil god Dameron was wreaking havoc across the universe and only a latter day Hercules could change past history and so affect the present and future. So it is that you are required to battle through five periods of ancient history; Greek, Roman, Norse, Viking and Egyptian before finally confronting Dameron himself. The game itself is a mixture of platform action and arcade adventure. Our hero must battle through a series of levels, acquiring different weapons as he goes. A quick tour of the first level will give you the gist of the game, the rest being variations on a theme.

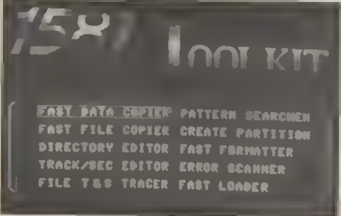
Starting off armed only with your fists, you must first thump one of the flying imps. He will drop ye olde flame thrower allowing you to throw a limited number of fireballs at future enemies. (Further hits on the imps will restore your health). Next, it's a case of zapping a few skeletons until one of them drops a sword. You can then use this to cut down a skeleton in chains. Follow him down to the bottom level, kill him with your sword so that his head falls into a fiery chasm. This will release a demon. Kill him with the fireballs, pick up the trident he drops and use that to kill the huge creature at the end of the first level.

That is really all there is to the game. It is simply a case of finding a weapon and using it in the right place. I must admit to being decidedly underwhelmed by the game, especially after all the hype and 'rave reviews' about it. I didn't find the screens particularly exciting and the animation and graphics were less than inspiring. I rather suspect that this sort of game works better on a Spectrum than it does on the 64. The game is a multi load and it takes for ever to load in after you die. Still, other people seem to like it so you may do so too. Though what everyone else sees in it remains something of a 'mythtery'.

AT A GLANCE
Title: Myth
Supplier: System 3
Price £9 99 (cassette) £14 99 (disk)

THE 1581 TOOLKIT





We put a utility program for the new 1581 disk drive through the mill
By Paul Traynor

The 1581 Toolkit is a collection of 10 disk utilities, but it is not just this as the package also includes the 1581 DOS Reference Guide and some other utility programs including Kracker-Mon. The Toolkit is for the 64 but will also support 64k VDC RAM or 1750 RAM expansion when used in 64 mode of a 128.

When the Toolkit is booted, after showing you a nice picture of two 1581 drives, you are presented with the initial menu showing the 10 modules to choose from. (pressing the space bar will allow you to toggle between the menu and the picture of the two drives) Once you have booted the Toolkit, subsequent loading of modules or reboots of the main system will be from the same device number as initially used. The first module is the Fast Datacopier which will work with either one or two 1581 drives, when you are using only one 1581 and a 128 this module will support the 64k VDC (for 7 pass copying) or 1750 RAM expansion (for 2 pass copying) to speed up operation time. In my tests a full disk copy using one drive and a 1750 RAM expansion took approximately 5.5 minutes. When you start the copying procedure the source disk is checked and if it is not write-protected the program will pause and tell you so. When copying is complete the program reports any read or write errors that may have occurred. The disk command option will allow the sending of any command to the drive that would normally require the 'open 15,8,15,' type of statement, examples of these are validate, new and collect. This ability to send disk commands is available in all of the modules. Other options on the Fast Datacopier menu are selecting first and last tracks for partial disk copies and also Reboot main menu. The next module is the Fast File copier and it is here that we find the program's first bad point and it is very important because although this file copier will work with one or two drivers and transfer from 1541/71 to 1581 and vice versa it will not have anything to do with relative files which is very sad if your database, like most, produce such files which you may need transferring. Support is provided for 1581 sub-directories, softwiring of drives (ie changing the device number temporarily), and VDC and system RAM expansions.

You also have the ability to format the destination disk before file copying. If you are using a 1541/71 you can alter the skew rate, this is fully explained in the manual but it has not been too successful in my tests As well as the reboot main menu selection you can also opt for exiting to BASIC The copying procedure produces an interesting screen display in the form of a bar chart The vertical scale represents the size of the file and the bar charts grow one at a time as each file is loaded and subsequently shrink again as each file is saved to the destination disk, the screen scrolls horizontally to accommodate any number of files One useful feature of the filecopier is you can move files from one sub-directory to another on the same 1581 disk, this is done by selecting both source and destination drives as the same number but choosing the required sub-directories. If you try to copy a file onto a directory or sub-directory where it already exists you will be given the option to choose another name for this second file

The third module is a Directory Editor, after loading the directory and selecting the edit directory option you are presented with the edit screen. This screen has space for an input buffer, where your directory will be at first, and an output buffer which shows the directory which you can write back onto the disk. Using cursor keys you can move around within either buffer or between buffers changing the positions of files As you point at each particular file it's information is shown at the top of the screen. This information includes the filename, type (rel, seq etc) and status (locked or unlocked), all of this information can be altered You can sort files alphabetically and numerically and also insert dividing lines into your directory

Other options include the ability to change the main disk title and id Module four is a Track and Sector Editor which has the facility to edit your data in a number of different ways, these are ASCII code, assembly language or in hexadecimal format, data used in the last two editing modes can be printed out on paper The editor has all the necessary scanning commands ie next or previous track, sector or link. The Track and Sector Tracer, module five, allows you to examine and change the contents of a file block by block This editing can be done in either disassembly mode or Hex/ASCII mode After reading the directory and selecting the file you wish to work on, you will be presented with a grid which shows a representation of how the file is stored on disk, ie you can see which blocks are allocated to it, using the cursor keys you move within these blocks and select one for editing in the modes outlined above

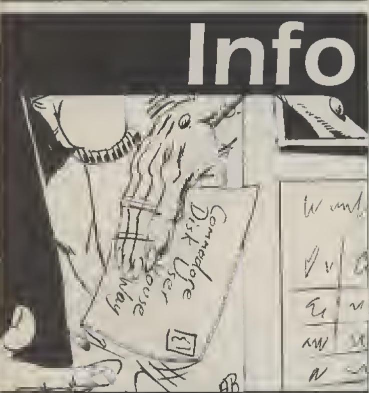
Module six, a Pattern Searcher or Fast Data Scanner, this module will search any number of tracks up to a whole disk for data which you can enter in three different forms Hex, decimal or ASCII or a combination of these. The number of matches found will be displayed and then you will be shown a grid which indicates which blocks on the disk contain these matches. You can display and edit information in disassembly or Hex/ ASCII mode and whilst in these modes any match from the previous scan will be highlighted Module seven, the Partition Creator, is described in the manual as fun to use and it is The approach used to manipulate your partitions is very interesting. Across the top of the edit screen are the tracks from 1 to 80 and partitions if any, will be shown hanging down from the track where they commence Your movement across the screen is restricted to the tracks taken up by the level you are at, this level is displayed at the bottom of the screen together with the options; create, delete and read partition. Although this may sound difficult it is very easy to get used to and extremely useful if you use multiple levels of partitions and sub-directories Module eight is a Fast Formatter, when I tested it, it took 20 seconds less to format a disk than the standard 128 header command It is also possible to select a range of tracks for a partial format of your 1581 disk Module nine is an Error Scanner which will determine whether a disk contains DOS read or write errors The results can be dumped to printer or displayed on four screens which are switched between by using the keys 1-4 These results will show whether or not a block has been used If a block contains an error an indication is also given as to which error is present. You are given the option to change start and end sectors for partial disk scans. The final module, number ten, is a Re-Locatable Fast Loader which boasts 900% faster loading of program files which in my short tests I found to be true! The fastloader can execute from different memory locations to suit the user, and the necessary files can be saved onto any work disk 1541/71 or 81 format but is only compatible with the 1581

As already mentioned the 1581 Toolkit package also contains the 1581 DOS Reference guide which is nearly 150 pages of text describing the make-up and operation of the 1581 All the information mentioned in the user's guide is expanded in this publication which covers topics including the new disk commands available, auto loaders, utility loaders, the disk format structure, burst commands and burst utility commands The aspects of hardware described includes the track cache buffer, the serial bus specifications, standard, fast and burst serial bus protocol and 1581 bugs The bulk of the book is the memory map which is split into three main areas; firstly 8k of RAM (which includes zero page work area, job queue, the data buffers and the BAM), secondly the DOS controller routines (which includes 8250A and WD1770/2 information) and thirdly the ROM memory map The documentation also outlines the use of the additional utilities which are included in the package, the main one of these is Kracker Mon which is a machine language monitor. Other utilities are an error creator, a diagnostic program for checking hardware errors, a method of changing suitable program files to user files, example utility command and auto boot files and a program which dumps disk drive ROM to disk for exploration with Kracker Mon.

The package is not cheap but does represent good value for money, because, as I have already mentioned, it is more than just a disk utility program The support for VDC and system RAM expansion means this package is worth considering for 128 users as well as 64er's. The very thing for those who appreciate the extra disk space and pleasing quiet operation that the 1581 drive offers.

Suppliers: F S S.L. Masons Rhyde, Defford Road, Pershore, Worcs
Telephone: 0386-553153
Price: £34 95

Techno Info



Despite its sometimes lengthy appearance, Techno-Info proves as popular as ever. Jason answers a few more letters.

Dear CDU,
I recently ordered a back issue of CDU which I have just received. One of the programs on the disk is CDU MENU KIT. I have used this and found that after I have made up the menu with my selected colours etc. and saved it onto my disk, when I re-load my menu the "Commodore Disk User" logo is the wrong colour and the moving striped effect doesn't work. At first I thought it may be a fault with my computer but I have tried it on my friends and it still does not work. I have also loaded your menus from other CDU disks and the logo on those are the wrong colours as well. I have found that if I load "Directories Explained" (Vol. 3 No. 4) then load and read through the Main Demonstration, when the computer comes to the end, if I then load my menu without resetting the computer the striped effect and the colours are fine. Could you please tell me if there is a bug in the CDU MENU KIT program and if so what I have to do to correct it.
R. Barker, Northampton

Dear Mr Baker,
Like the problems with the game Wabbit that I discussed in the April issue, this problem is caused by the way older models of the Commodore 64 clear the screen. There is no bug in the program. With the older models, when the screen is cleared the computer POKEs values into the colour RAM area dependent upon the background colour at the time. With the newer models, the values stored depend upon the cursor colour. The colour rolling techniques used by the menus relies on multicolour characters with colour values of between 8 and 15. On power-up the screen is colour 6 and so this is the colour that the characters assume on older models and so the characters are not multicolour. Hence the effect is not obtained. I must say that I am glad you found the "Directories Explained" program of use, even if it did only serve to allow you to see the colour rolling effect. This is because it exits with background colour 11. To rectify the fault you will need to POKE53281,15 each time before you load the menu. However, after a while this will become tedious and so on this issue's disc you will find the file "PROBE1". Load and run it as a BASIC program and then supply the name of the pre-created menu and the new one that will be created and will work. Insert the correct disk and then wait. The computer reads from the old file and writes to a new file, changing two significant bytes where needed. The whole process takes a while so be patient. The piece that is altered changes the cursor colour prior to the screen clear. The new version will change the background colour instead. Of course this only works on the old models that don't usually produce the colour rolling. To all programmers out there I send the plea that you must remember that old 64s POKE colour differently when clearing the screen. I hope this rou-

tine will be of use to everyone who owns the old models.

Dear CDU,
I am currently creating a demo which uses a high resolution screen combined with sprites to give the effect of movement. The demo is of a face, speaking fluently with the help of the sprites as mouth movement. The problems started when I couldn't find a single sound sampler on the shelves at a reasonable price. That is when I wrote to you, hoping that you will be able to recommend a relatively inexpensive sampler. That is if it is possible to have sound sampled speech continually speaking at the same rhythm as the mouth in the demo. Maybe you could continually load in speech as it is executed or load in a long string of speech to be played and updated at intervals. If you could answer this query I would be most grateful.
Lee Bamber, Wigan.

Dear Lee,
I am afraid that you won't find a decent sound sampler on the market at a lower price than about fifty pounds, unless you want a synthesised speech program or separate module to produce the speech – but then only people with the same equipment will be able to use your demo. Unfortunately there is more bad news. Digital Sound Samplers eat away at the memory like nothing on earth. I have a nine second sample of a song and it occupies over 20K. The timings required for a decent output are very precise and you certainly wouldn't be able to keep loading the speech, unless there were long pauses. Sorry to disappoint you but if you do want to make a first class job you will have to save your pennies and have only about ten seconds worth of speech.

Dear CDU,
I have just two queries. I am moving to Canada and would like to know what modifications I would have to make to my system. Do I have to change the voltage and if so how do I go about this? Secondly, I own a 1520 plotter but one of the pens has run out and no dealer I know sells these pens, can you help me here?
Yasin Poptani, Middlesex

Dear Yasin,
I would think that the only thing that would need to be changed is the power unit of the 64. You will need one that has an input of the Canadian voltage but one that keeps the standard output for the 64. However, I would talk to a qualified electrician on that who could tell you what exactly it entails. Your second query I can be a little more definite on! Last year I obtained a new set of pens from HRS Electronics. Their address is Garretts Green Lane, Birmingham B33 0UE and the telephone number is 021-789-7575. Other than that company there are a few Tandy shops that sell very similar pens because the Tandy plotter is almost identical.

Dear CDU,
I own a Commodore 64 and 1541 disk drive, with which I am now experiencing problems. I can use the disk drive with no problems for months and then suddenly the drive comes up with an error number 27 – a read error. The program will either not load or the drive working light stays on continuously and the computer will not reset. Sometimes the directory will load and when viewed the filenames are changed to graphics characters. If the drive is left for a few weeks it will sometimes load normally. At other times, the disk has to be reformatted. I find that most disks will then work with no problems. I hope that you will be able to help with either information on how to cure the problem or how to uncorrupt my disks. The computer is approximately six years old and the 1541 about eight.
G. Collins, Surrey

Dear Mr. Collins,
From what you have said I would think that the problems lie not with the disks but with the drive itself. I cannot see how a disk can work one minute, not work a few months later and then work again at some future date if the disk is corrupt. I should think, and without trying one of your disks I can't be sure, that the drive is not aligned correctly. This is quite possible after eight years of use. You may benefit from trying one of your disks that does not work on a friend's drive, if this is possible. Or you could take it to a computer shop and ask them to check the head alignment and motor speed which should be in the region of 300rpm. The reason that disks will sometimes work after reformatting is because if the head has become stuck, the "knocking" that occurs at the start of the format can correct the head position. As I say, get the 1541 checked out, or possibly buy a new drive.

Dear CDU,
I recently acquired an EPSON RX80 printer and set about connecting it to the user port of my C128D which I use in the C64 mode most of the time. The connection to the printer is via approximately two metres of ribbon cable. There is nothing wrong with the RX80 – it works a treat. But I have found that when the RX80 is connected to the user port I cannot load some of my software from disk. For example the CDU menu produces about ten lines of garbled ASCII and with another of my programs the drive runs forever but the title screen never appears.. Even with the RX80 powered down but still connected the problems persist. The only solution I have is to leave the RX80 out of circuit completely and then all my software loads fine. The peculiar thing is that some software will load and run perfectly with the RX80 in circuit and powered up. Can you offer any advice as to the possible cause of the problem and how I can keep my RX80 in circuit at all times?
Ian Stewart, Aberdeen

Dear Ian,
The problem and symptoms sound very similar to the sort that can be experienced when fastload and backup cartridges are in place. There are some connections between the user port and the expansion slot of the computer, for both voltage and linking addresses. This could have something to do with the problems that you are experiencing. There may also be a daisychaining problem although this is unlikely. Other than to

do what you are doing at present, I can't really offer any more advice as to how you should rectify the problem. Still, I shall delve a bit further for you and report back next month if I find anything worthwhile

Dear CDU,
I have just three questions to ask you. Firstly, is it true about a laptop 64 and if so how much would it cost? Secondly, is the machine code equivalent of the BASIC load command just as simple and effective. What I mean is, would the program automatically run if given a start address in the load program. And finally, how on earth do I put character sets designed on a character editor such as the "Ultimate Font Editor" into my working machine code programs?
Stuart Smith, Manchester

Dear Stuart,
The answer to your first question is 'probably'. Unfortunately I cannot supply you with a more precise answer as I have only seen it mentioned, as I presume you did, in the FSSL catalogue. I have no idea of the sort of price that one is looking at but I can see no reason for laptop 64 not to emerge at some future date. The machine code LOAD is just as effective as the BASIC one and in fact I prefer the machine code version because there is no need for variables to keep track of where your program should jump back to once the correct piece has loaded into the machine. The full description can be found on page 286 of the Programmer's Reference Guide. But on this issue's disk you will find an assembly language source file, "PROB2", just in case you do not own a book outlining the method to use. It may seem a little more complicated than it really is at first though. Now to your last query. You save the character set out using a built-in option and then to use it in your programs you load it back (with the ,8,1 suffix if loading from BASIC) and then simply POKE the correct value into location 53272. This can be found using a number of methods, but for character set number one saved by the Ultimate Font Editor the procedure is as follows. Load in the set in the usual way from your program and then, because the information is stored $2000-$2800, you issue the command POKE53272,24 (LDA #$18, STA $D018 in machine code). Your character set will make itself known!

Dear CDU,
I have been wanting a program that would enable me to print pictures drawn using CDU Paint to my MPS801. I carefully followed the instructions in CDU Paint and in the program "Colour Picture Printing" and the enclosed printout is the result. I need hardly say that it isn't what I wanted! However, on checking to find out where I went wrong, I found what seems to be an inconsistency in the "Colour Printing" program. In the magazine the instruction is to type SYS34840 but the screen prompt says type SYS35840 and this does activate the printer. But, try as I might, I cannot seem to get the program to print out what is on file. It loads, comes up on the screen, but will not print. Can you please tell me where I am going wrong?
Rev. W. Farquhar, Dunfermline

Dear Rev. Farquhar,
I am glad you were able to sort out the problem with the start address. However, you have confused the purpose of the program. The program in question is concerned with reproducing colour pictures in colour on the paper. Obviously the MPS801 is only a black and white printer – thus no colour is obtained. The reason for the mess that you got lies in the fact that the control codes that are sent to a colour printer will do something, whereas sent to an MPS801 they won't. What I suspect you want is actually a program that prints colour pictures in shades of grey. You could also purchase a plug-in cartridge. The "Power Cartridge" from BDL of 89 Bewick Road, Gateshead, Tyne and Wear costs only seventeen pounds and would merely entail you displaying your creation and then pressing a button followed by selecting an option from a menu. As I said, the program in CDU was only designed to operate with printers that can output colours on to the actual paper. I hope that I have been able to clarify the problem.

Dear CDU,
I have a Commodore 128D to which I would like to link a Commodore 8050 double disk drive. I have an Interpod together with all the necessary leads, but I have not got an inkling of the protocol needed to address the 8050, could you help? Perhaps one of your many readers has a spare manual that is no longer needed that he or she would be willing to sell me. Well it's a thought anyway. Do you know of a program that will enable me to drive a Star LC10C printer from the user port on the 128D? I would be more than grateful for any information that you can give me on this as it does seem a shame to have this piece of equipment lying idle.
Bert Richardson, Middlesex

Dear Bert,
Well, has anyone got a spare manual? If so, please write to Techno Info supplying your name and address and price that you would want to part with it. So far as I know there is no way to connect and operate a STAR LC10C printer from the said user port, although I must confess I am not sure (slap wrist!!). If anybody out there knows different – please come forward with the information!

Dear CDU,
I am having a problem with the BASE-ED program featured on one of the Commodore Disk User disks. I am using high quality disks and so it is not those that are at fault. My friend also experienced the same sort of problem when using the program. If you can help in any way to alleviate the problem I would be most grateful. I shall run through exactly what happens when I use the program. Once the disk has been formatted and the blocks allocated and the fields set up, everything is all right. However, when I try to add a record with number 452 and type the information into the field,s when it comes to saving the information I get an error number 66, illegal track or sector. Pressing RETURN takes you back to the Record Manipulation menu and from there using 'read record' I can call up record 452. However, this must be

done as part of a range of records as the 'read one only' option does not seem to function correctly either. The information appears but after exiting the program and reloading the file, when I read in record 452, each field is marked as empty, indicating that the information has not been saved. Please could you point me in the right direction.
H. Coughlan, Cumbernauld.

Dear Mr Coughlan,
Unfortunately an error seems to have crept into the program, one which was not rectified with the publication of BASE-ED2 Before telling you how to cure the problem, for which you will need an external reset switch as part of a backup cartridge that has an OLD command built-in, I will send out another plea to programmers. It is simply not possible to test all programs to destruction here and so please ensure that your programs work thoroughly before submitting them Back to the problem at hand, If you do not own such a device then I am afraid there is no simple way to cure the fault – perhaps CDU will publish an amendment program at a later date – I shall put it to Paul, el Ed. If you do own such a cartridge then get the program loaded and running. Press the RESET button and then type OLD to recover the program Unfortunately it has been compressed so you cannot perform this operation until you have RUN the program and the author has put a few POKEs in to disable the STOP key which doesn't help matters Now LIST line 2430 and change the line number after the THEN command to 2590 Next LIST line 4010 and change the command F3=25 to read F3=19 That will rectify the fault with the sector numbers. You will also find that with entering enough records the program would have originally written information over the records and so you should make this alteration if possible Then you need to retype the following lines as shown:

```
10 DN = PEEK (186) POKE53280, 0:
POKE53281,0:POKE646,15.RL=254,
CS$=CHR(147)
20 CR$=CHR$(13) CD$=CHR$(17)·
CU$=CHR$(145) CL$=CHR$(157)·
RC$=CHR$(29)
30 DIM I$(500), C$(30)FB$(30),
L(30).FL%(30).FM% (30): OPEN 15,
DN,15
SYS49152
```

Then type RUN The program will pause for a few moments and then you will be greeted with the main menu Then press the button on your cartridge to allow you to save a backup of the memory This will then create a working copy The reason you can't just save it normally is because there is some machine code However, this could be saved independently if you find the start and end addresses The main program could then be saved with the standard SAVE command I hope that you will be able to rectify the fault using the above technique.

Dear CDU,
I would be very grateful if you could put the following humble plea in your letters column: HEEELLLPPP!!! Has anyone out there got a copy of "Laser BASIC" for the 64 for sale as I cannot find it anywhere. I am willing to pay any price (almost!). Thank you.
Simon Searle, Derbyshire

Dear Simon,
Thank you for you letter I hope that one of our dedicated readers can sell it to you at a fair price. Of course, it must be the original and not a pirate copy! So if any of you are able to help, please write to Techno Info with your name, address and price that you want, and then I shall put the person who's offer is the lowest (or the first that is reasonably priced!) in touch with Simon

Dear CDU,
I have recently had some excellent service which I feel worthy of note to you and your readers. On the first of February this year I ordered a 1541 disk drive and a Star LC10C printer from a company called COMPOST. They duly arrived the next Monday, the fifth. Unfortunately after testing the printer that evening I found it to be faulty, I contacted COMPOST the next morning – "No problem, Sir, We will exchange it for you." And sure enough, within two hours there was a van outside my house to collect it. Two days later, the eighth, the same van pulled up at my home with a new printer that works perfectly. So COMPOST, take a bow, you surely deserve one for that really first class service. I for one will be only too pleased to do more business with you in the future!
R. J. Ayers, Mansfield

Dear Mr. Ayers,
Thank you very much for informing us of this company. I would agree – COMPOST take a bow. My local computer shop should follow their example For anyone who wants a sample of such wonderful service, Mr. Ayers has very kindly supplied the address and telephone number. You can contact COMPOST at Unit 6, Forest Close, Ebblake Industrial Estate, Verwood, Dorset BH21 6DA or you could telephone them on 0202-292195

Tip of the Month

We still haven t had many of you sending us your general tips to show off your obvious intellect – well you do read CDU after all! So I shall present you with something that I hope you will find of use. It is a very short machine code routine that will allow you to SAVE an area of memory, including the RAM underneath the interpreter ROM ($A000-$BFFF), without the need to change every pointer under the sun It is on this issue s disk, titled as TECHNO TIP Load it using the ,8,1 suffix and then whenever you wish to use it type SYS679 F$,d,2,s,e where F$ is the name of the file, d represents the device number, s' the start address of the piece of memory and e' is the end address plus one For example, to save the code itself to your own disk you would type SYS679"MC SAVER ,8,2 679,734 It is as simple as that!

However, I would much rather be able to present something by one of you (no matter what it is POKES, routines, anything goes – almost!) So get your tips flooding in to Techno Info, Commodore Disk User Argus House, Boundary Way, Hemel Hempstead, Herts HP2 7ST That is also the address to which you send details of any programming problems that you are encountering See you again next month!

Noddy's Revenge

We ask the question is this the result of too much computer violence?

As told by PC Plod

Having read a report regarding the controversial subject of restricting the purchase of "violent" games (such as Rambo, Operation Wolf, and even the mainly text adventure Jack the Ripper) by issuing age certificates or even in some cases, banning the game completely, I decided to sit down and put pen to paper – or at least fingers to keyboard. In case you are unfamiliar with the subject, the argument is do you play these games to simply marvel at the excellent animation and because the fast action is so addictive, or do you play them to see the look of terror on a man's face as you fire a bullet or two into his skull? In the following story I am not conveying my personal views on the subject – this is neither the time nor the best place. Instead I shall leave it up to you to decide whether I have merely written a pointless story to partially emulate the gore of Stephen King or whether I have "fictionalised", if such a word exists!, the effects that one of these so-called "hack-'em-up' computer games could have upon somebody's mind. So it may be extreme, but is it possible with some people today? I don't know – the next cry will be that my mind has been warped by playing too many hack-'em-ups. Still, enough said already, so if you are eighteen years of age or over please read on and let Noddy's Revenge, the tale of the Camberwick Green Massacre, begin!..

Everything was quiet on Camberwick Green, except, that is, for the whispering of the few trees as the wind rustled their leaves. In a small house just a few yards from the Post Office, there came a stuttered banging noise. It was emanating from Noddy's house and was so profound that all his little yellow model cars on the lounge shelf were rattling up and down. In the background one could hear the title tune from the computer game Rambo 47. Noddy had only recently purchased the Commodore 64 from a local shop and enjoyed playing these games because they were so fast and addictive. The light came through a small window, the evening sun reflecting off the monitor screen. Noddy raised the axe once more and brought it down on the now rather tattered remains of Andy Pandy. "Why... won't... the... stupid... thing... fit..." puffed Noddy to himself as the axe came down once more.

There was a small pile of earth and a hole next to it which Noddy had hidden with the 64's dust cover. Andy Pandy, gullible nurd that he was, didn't quite fit in the hole that Noddy had prepared and so instead of making it wider, which would have called for extra time and effort, Noddy had decided to hack Andy Pandy's legs off. "A much more sensible idea," thought Noddy, as he hummed a little song to himself, cutting through the last tendon in the process. He then rammed what was left of the body into the hole, concealing it again with the dust cover. The rest was thrown uncanningly onto the open fire that warmed the otherwise cold house of Noddy. "Always so gullible", thought Noddy, "always letting that stupid woman boss him about and make him search for Looby Loo in that cutesy high-pitched voice!"

You see, Noddy had waited all afternoon for Andy Pandy to come round. Noddy was gazing blankly into the open fire, humming a little tuneless song to himself whilst fondling the oaken handle of the axe. After a while, though, Noddy thought that Andy would never appear and so he put the axe back into the cupboard. However, when Andy Pandy had finally turned up, knocking on Noddy's door as if he were held up by strings, Noddy had answered the door quite normally, talked about the wonderful Camberwick weather, and then had gone to the cupboard and fetched the axe. Whilst Andy Pandy had sat there, saying nothing but with a painted expression on his face (which Noddy only served to increase) Noddy had raised the axe high and brought it down with a resounding thud on Andy Pandy's skull, cleaving it virtually in two, with just some muscle fibre in place to keep the pieces upright, and freezing Andy's eyes wide in horror that Noddy could do such an awful thing! He giggled a little as he remembered that delightful sequence from one of his new computer games that seemed to be an action replay of what had just happened. He had actually done it in reality now and was so proud, wiping some saliva from his mouth. Noddy had then mopped up the blood, washed the axe in case it were to require further use, and then as if nothing had happened, Noddy had begun to dig the hole.

Big Ears wondered why Noddy had not been around the town in his bright yellow car that morning and so he decided to visit Noddy instead. He admired the evening sun, and listened to the birds singing. Noddy looked out of his window and saw Big Ears coming closer and closer. Although it meant losing his super-high score on Rambo 47, Noddy unplugged the 64 and replaced it with his drill. "What's a stupid high score when I can get the same satisfaction without playing the game," thought Noddy to himself.

Big Ears had no time to realise what had happened, for the moment that he had opened Noddy's door and stepped inside, Noddy had slammed it shut and then proceeded with the drill. Piercing Big Ears' skull and sending a beautiful fountain of blood all over Noddy's face and jester's hat. "How wonderful!" he cried, as he jumped up and down with excitement – "this never happens on any of my stupid computer games!" Noddy then pulled Big Ears further inside and put him in the cupboard with the axe that had killed Andy Pandy.

Re-installing the computer, Noddy loaded up Operation Wolf. He had never bothered to read the instructions of any of these games. What's the point, anyway? Noddy enjoyed playing this particular game more than anything else because you actually looked face to face at who you were killing, which Noddy thought was a wonderful touch of realism. After half an hour, Noddy finally got blasted himself. Still, he experienced a euphoric feeling of power that made him even more irritable. Andy Pandy and Big Ears had definitely deserved everything that they got. When night had duly fallen, Noddy dragged the bodies outside and buried them in a make-shift grave by the side of his house. "Auf Wiedersehen 'friends'," Noddy giggled, "things are going to change round this town now that I am in charge." He laughed hysterically and went back indoors.

Morning came and Teddy and Looby Loo made their way to Noddy's house to see whether he knew where Andy Pandy and Big Ears were because nobody had caught sight nor sound of them since yesterday afternoon. Wasn't that a surprise? When they reached Noddy's house the door was wide open and Noddy was nowhere to be seen. Teddy and Looby Loo looked inside the house and noticed a hole in Noddy's floor. The monitor was still on and a notice was stuck to the wall with a blob of congealing jam that Paddington had given to him about six months ago. It read, "out getting the hefflump". "That's strange," thought Teddy, "there are no heffalumps around Camberwick Green, only flowerpot men. What is Noddy up to now?"

Not even the partially warped mind of Teddy could have imagined what Noddy was up to. That morning he had woken up with a splitting headache after having a dream about what he had done to Andy Pandy. To take his mind off the headache he decided to sit down at his computer and have another blast at Rambo 47. It seemed to give him another brilliant idea. He left his house with a plastic container with the word INSECTICIDE written on the side and DANGER below it in large red letters. He took the container into the middle of a field, in fact to Muffin the Mule's favourite patch of long grass. "This will serve that poor excuse for a horse right," laughed Noddy loudly as he sprayed it over the grass. "Always claiming that he could beat me at my computer games." Noddy then hid behind a decidedly large tree and waited for Muffin the Mule to approach. About twenty minutes later he came trotting over the hill and Noddy watched the unsuspecting Muffin eat himself to death – sheer poetic justice after he had spent five hours playing an adventure, only to die of food poisoning from a sandwich that was found in a cupboard. Noddy dumped the nearly dead body of Muffin the Mule in the same grave as Andy Pandy and Big Ears. Muffin the Mule's eyes seemed to be staring in disbelief – "You are very lucky I didn't chop you into little bits and then feed you to Teddy!". Noddy continued to laugh manically before once again concealing the make-shift grave with the 64's dust cover. Talk about dual purpose goods!

Noddy didn't return to his house until dinner time and by that time he was missing his computer very much. So when he returned to his house he was in quite a bad mood and all he needed to make him absolutely mad was the sight of Teddy and Looby Loo jumping up and down outside his house singing "Bouncy, bouncy. Bouncy, bouncy". Looby Loo was holding a large parcel with "Do not bend – contains computer software" written on it in big letters. It probably wouldn't have made any difference to Postman Pat. He usually bends things just to be spiteful – the old fuddy duddy – and as for his cat!! Inside the parcel was a load of software that a friend of Noddy's in Trumpton had copied illegally for him. "Ha! Ha!", thought Noddy, "all those authors that are getting ripped off because of piracy." This lightened Noddy's mood somewhat.

Noddy then felt extremely lucky when Looby Loo said that she had to go home to play with her building blocks, because that left Teddy at his mercy. Noddy smiled slyly as an idea began to form in his over-active brain when Teddy suggested they go and play that game that Pooh Bear had showed them how to play – which they called Pooh sticks. "What a great opportunity", whispered Noddy to himself as he followed the innocent Teddy to the bridge over the River Turn.

Once they were on the bridge and the rather pointless and infuriatingly boring game of Pooh Sticks was underway, Noddy thought how he would much rather shove his stick somewhere that Teddy would find rather painful instead of throwing it mindlessly into the fast flowing stream beneath them. Teddy was leaning over the side of the bridge looking for his stick and so did not see Noddy's wide, horrific grin as he outstretched his arms and moved towards Teddy with the cruel intent of pushing the stupid thing into the stream. "Oh what fun this will be," thought Noddy, "just like that scene on the Last Ninja where you push that twit off the cliff!"

There was a loud splash as Teddy hit the water and started to struggle as his head was slowly covered by the water. He was gulping and choking. Noddy was holding on to the rail of the bridge and jumping up and down with fulfilment, joyously shouting at the drowning Teddy. "Why?", spluttered Teddy sadly as he slowly started to turn blue with the cold of the water that was engulfing him. Noddy found this somewhat hilarious. "I'll tell you why you stupid imbecile!", "It serves you right for making jokes about my red mudguards!" Teddy was unable to hear this answer as he had already floated quite a long way downstream. He was face down – dead. "Good riddance" laughed Noddy as he looked at his watch. "And there is still time to get that pest Looby Loo. She should have played herself to sleep by now!" The sound of the bell on Noddy's hat became less audible as he strolled off into the distance.

Noddy sneaked through the garden gate and past the sleeping figure of Weed. Looby Loo was asleep on top of her large building blocks. You could hear the cogs clicking around in Noddy's head as he thought of the best plan. He remembered the film "The Day of the Triffids" and although it didn't arouse as much satisfaction in him as it would have done had he been able to duplicate a computer adaptation, he carefully picked up Looby Loo and took her outside It was there that he laid her down as he pondered whether there was another option – perhaps he could blast her head off with a sub-machine-gun like in Robocop After all, emulating what happens in a computer game is much more enjoyable because with a game you come so close – you actually kill the people yourself.

Still, he settled for his original idea and carefully, with Looby Loo in one arm, lifted up Weed by her stem. He laid Looby Loo at the bottom of the flowerpot and then replaced Weed Almost immediately he heard the muffled screaming of Looby Loo as she was gradually suffocated by the weight of the mud on top of her Weed slept through it all, good job as well – for if she hadn't, Noddy would have certainly pruned her so much that she wouldn't grow again next summer! Noddy skipped back home, smiling at the thought of Looby Loo finally having her mouth filled up permanently

The computer was still on and Noddy sat down to play a final game of Rambo 47 before he tried out his new software that had been sent to him Five . ten. .twenty . thirty. forty minutes passed before he finally gave up. He was getting annoyed because killing people on the computer was becoming boring. He had been able to go one step further and actually kill the people in reality in almost the same way that he would have on one of his computer games. The thought of how he would one day get round to Postman Pat had suddenly entered his head as he stuck a knife into the stomach of one of Rambo s enemies a couple of minutes ago 'No, that would be too good for that wretched postman,' he said aloud as he unplugged his 64. There was a knock on the front door – 'Post for you, Noddy" It was a good job that Noddy had become bored with his game for when he heard that come through the rotting front door he immediately picked up and plugged in his new chain saw .

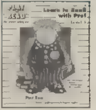
Well, I think that you can figure out what happened next. If you laughed or even smiled at all whilst reading that story then maybe you should have a serious think, and then perhaps laws of censorship and age restrictions on computer games should be more rigidly enforced But hang on a minute – were you laughing at the fact that those creatures were being cruelly killed or were you amused by the thought of such an innocent children's character performing those deeds? All right, I have exaggerated the storyline by using Noddy but it just serves to show that this type of story can have any sort of basis It is totally unoriginal, very predictable and contains the same mindless and senseless acts as those in the shoot-'em-ups that we all either love or hate, and even though I wrote it, I would agree that it becomes monotonous after a while – but you must already read on to find out what happens next Does it sound a bit like what goes through your mind when you play those shoot- em-ups? I think it does There are also the role playing games – like Dungeons and Dragons – that some people think are the basis of witchcraft and devil worship Where on earth SHOULD the line be drawn? I shall leave you to decide

Commodore DISK·USER·

Readers Offer

Learn to Read with Prof

Part 1: Prof plays a new game

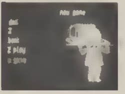


Here's a great CDU and Prisma Software reader offer to save you £££ Your child can have the chance to combine reading with fun With the help of 'Learn to read with Prof' your child can play a super computer game and learn to read at the same time

Disk version £17.99 M.R.P. £22.99

Tape version £14.99 M.R.P. £19.99

Learn to read with Prof has three stages to success Children first of all listen to a story on cassette read by the lively actress Patricia Hayes Stage two is where the colourful character called Prof helps your child to read a group of words in a 'computer game' In stage three after each group of words is learnt your child is then able to read one of the FIVE books



Learn to read with Prof' has taken over 4 years to design with the help of hundreds of young children across the U K Positive attitudes toward reading and learning are formed in the home during the early years so start them off with a love for reading

Telephone orders 0442 66551

Everything for your budding professor well almost.

1st in a series of 12

Please supply . OCDU/1 @ £17 99 Total . . Please supply ... OCDU/2 @ £14 99 Total £ ...

Cheques/P O payable to **A.S.P.** or debit my Access/Visa

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Expiry . . . ...

Signature . . . . Name . ...

Address . . . .

Coupon to CDU, Reader Offer, Argus House Boundary Way, Hemel Hempstead, Herts HP2 7ST
Allow 28 days for delivery (U K only - overseas upon request)

Teach reading with 'Play and Read' and be confident of success !

CDU Reader Offer, Argus House, Boundary Way, Hemel Hempstead, Herts HP2 7ST

Thankyou for your order

'C' Here

we start a series looking at the possibility of programming in 'C' as the alternative language

By John Simpson

If you want to look out to C, you will find it really is worth your while. It is a fact that most people who program computers to do their bidding invariably start off by grappling the intricacies of BASIC, and then, if they are really keen to construct fast action, smooth running, and totally controlled programs they may move on to ASSEMBLER

BASIC is, as it says – basic (although this is not what BASIC means! It is an acronym for Beginner's All-purpose Symbolic Instruction Code). However, it is reasonably easy to use and is not too difficult to learn For the Commodore 64's version of BASIC there are 70 or so 'keywords' which need to be committed to memory, and then quite a lot of juggling with the commands needs to be accomplished to achieve reasonable results – as an example the IF· THEN. ELSE statements, Else' in the BASIC which is delivered with the C64 doesn't exist, however, with some juggling, which regrettably, uses* up more memory and does slow down execution time somewhat, can be simulated.

One answer, of course, is to purchase a 'BASIC Extension' type of program... several of which have been published in Argus Publications from time to time. However, this does mean extending your 'keyword' vocabulary, and sometimes by as much as a couple of hundred or more 'keywords'. Phew!

ASSEMBLER, on the other hand, is much easier to learn – there being only fifty-six mnemonics or code words in the instruction set of the C64 Using it, on the other hand, can be somewhat more difficult and a thorough understanding of the machine is of fundamental importance. Of course, once this has been achieved and your program has been assembled directly into binary machine-code, then nothing could be finer. It's as fast as light (!) being hundreds, and in some cases, thousands, of times faster than BASIC

The reason for this is that BASIC is, as you probably know, an interpreted language What happens in the murky depths of computer memory when you run your BASIC program is that every single byte on every program line is shunted off to a massive interpreter program where it is transformed into binary machine-code (and not always the most efficient MC at that!) and then executed, after which it returns back to the BASIC program to 'grab' the next 'chunk' of instructions to be interpreted Oh so slow, Sprites are only able to amble across the screen rather like jerks at a jelly party, and the transferring of blocks of data from one place to another tend to dribble along like a stream in a drought

In ASSEMBLER, on the other hand, the same sprites can easily traverse the screen so fast the eye perceives only a flash, and block data transfer is seemingly instantaneous Split screens, multiple sprites, and full multi-track musical scores all being possible within a single program. Witness today's modern games!

Of course you can get hold of a BASIC compiler. The difference between a compiler and an interpreter is the latter interprets the BASIC into machine-code whilst the program is actually running, and the former will compile each BASIC line of code into machine-code and finish up with a separate machine-code program which runs independent to the original BASIC program The compiler will take the BASIC program, and, after a great deal of time for a rather complicated and large program, translate it so you finish up with a rather clumsy machine-code program which is faster than the original BASIC, but only a touch or two!

So where is the middle ground? What is the language which lies somewhere between the ease and readability of high level BASIC and the speed and control of low level ASSEMBLER? The answer lies out to C.

Before I dive into the C language there is just one more point Portability The writing of a program on one machine and the running of it on a completely different one

A program written in BASIC V2 (Commodore 64 Standard) can, but only with a great deal of fuss 'n' bother – the rewriting of chunks of the program, changing keywords, possibly screen codes and memory addresses – be transported to another machine A task most programmers have no wish to do! ASSEMBLER, on the other hand? Very simply put.. no chance! It would need a complete rewrite using a totally different set of mnemonics for, say, the Z80 chip C? A completely different kettle of fish! But more of this later

C – how did it get such a simple name? Well if you are that interested, the early ancestor of C was a language developed at the Cambridge and London universities sometime around the early 1960s and which was eventually transformed into BCPL, a systems programming language which incorporated control structures needed for structured programming Later a chap called Ken Thompson, working at the Bell Laboratories, derived a sub-language from this into what became known as B. Then along came Dennis Ritchie who, in 1972, invented C, a name chosen to indicate a follow on to B The aim of C at that time was to write operating systems to work with the UNIX system which was still under development at that time.

So, a big thank you Dennis.

What are the advantages of C? I suppose they can be summed up as Structure, Portability and Compactness.

Structure, in BASIC, although not completely impossible is often not put into practice, usually because the learning of BASIC is often by experiment/accident trial and error and quite usually with GOTO commands splattered every which way

Naturally a program is only as good as its final function In other words, does it accomplish that for which it was originally intended.. and if it does, and if the program will never need revising, or updating, or another programmer taking over , then fine – but should it need one or the other of the above then the unstructured program can take tedious and frustrating gulps of a programmer's very valuable programming time just to sort things out and to discover just what section of code is intended to do just what.

A well structured program is usually truly modular in its form. This means that one module can be completely independent to other modules, and even to the main processing program. This is one of the major differences from BASIC to almost any other programming language. It is not possible to have a completely structured BASIC program simply because BASIC subroutines are not truly modular. As an example of this if we were to construct a subroutine with an integer variable, which we shall call DUMMY, within it, then altering the value of DUMMY anywhere else in the program will alter DUMMY in the subroutine (and elsewhere for that matter). This is known as a Global variable. Fine, but as a consequence of this the BASIC programmer must take great care when specifying variable names in subroutines, and will use memory space and time spent in the passing of values from one variable name to another.

In C the equivalent of a subroutine is known as a FUNCTION, and all of the variables in a function are completely local to that function. In other words if you assign a value to DUMMY within a function then it only exists within that particular function. Should you have another function, or even the main program, with a different DUMMY variable within it, then it will not have the same value, not any effect upon the first, or vice versa.

It is only by using deliberate programming that you are able to pass a value to a function or back again. This makes it an easier task to design modules separately and without the worry of variable names which may have been used in other modules or the main program. Thus rendering each module as a complete 'stand-alone' module and with total control over all of the input and output operations.

A really great feature of C is its portability, in other words a program written on one machine can, with recompiling on a different machine, be just as easily run on that different machine. There are a few conditions imposed but they are usually reasonably obvious. The compiler must allow for the normal C statements. Some of the smaller machine compilers may not include fractions, or floating point numbers. If a program includes screen effects, such as graphics, then it will need library functions (more of these a little later). Basically, then, it is the compatibility of compilers rather than machines that count for portability.

As far as compactness is concerned it is possible to write a program in C and then by making adjustments, and shortening it quite considerably. It can be written in such a way that gives faster running code, faster compiling time and is so much shorter and easier to read.

One quite astounding difference is the unusually small amount of reserved words within the language. As you probably know and as I mentioned somewhere earlier, the development of BASIC in recent years has been to create bigger interpreters adding possibly 150/200 or more extra reserved 'keywords'. All of which need added routines, system variables and vectors to execute them and in so doing using up very valuable RAM memory blocks. In reality C is rather a small language but which has been designed in such a way as to make it very powerful indeed.

All the actions you are most likely to need are contained in a functions library compiled and ready for your use. The functions library will contain all the routines you are likely to require. Such things as screen formatting, printer options, open and close files, and etc. A comprehensive 'library' will be included within the C package which you purchase. For example, "Spinnaker's" POWER C contains many more than a hundred plus library functions' these range from converting strings pointed to by argument into either integer or float quantity through system calls to machine code to general purpose sort routines. It does not stop there either, you can add new functions to your library as and when you desire.

Staying with POWER C for the Commodore 64 by "Spinnaker" let's look into more detail about just what you might expect from the C package. First, there are two disks (a) The System Disk and (b), The Functions Library Disk. The Functions Library Disk contains a vast library of all the different functions you will need to create your programs, but as time goes on and you become more proficient in the use of C you will find yourself adding to that library. Routines are functions to control Sprites, split-screens, hard-scrolls, sound-effects or whatever. On the System disk are the programs which you will need to edit your program, compile it into object code, trim it to optimise space, and link together different modules, or programs, and library functions.

After loading you enter the SHELL program which creates the operating environment from which the rest of the programs work. From the SHELL, you can call' the Editor program which comes with or without a syntax checker. It is within the Editor where you will write your program, make changes and in general edit until you are satisfied that it is ready for compiling into an object code (binary machine-code). So, back to the SHELL and a 'call' for the Compiler program. Once the program or module is compiled you may want to optimise the compiled object code to make it as small as possible, gaining valuable RAM, so from the SHELL a call for 'Trim'. Once this has run through the program you are then presented with a screen display of the results of its efforts on a percentage basis. At this stage you will want to link in library functions and/or other modules, so again, from within the SHELL, you can call up the Linker program which will take care of this task for you.

Once the linking is done the program will request a filename and save out the completed object code program to disk which, depending upon the suffix used, can then be run from either within the SHELL environment or as a stand alone program which can be run on any C64 without the SHELL operating environment.

Here follows a simple BASIC and equivalent C program to whet your appetite for the next article 'Let's Get Down To Coding in C', next issue. Doesn't look much different, does it?

```
            *BASIC*

10    PRINT, "HELLO EVERYONE"
20    END

            ***C***

main()
{
printf("\tHello Everyone\n");
}
```

To sum up, then, I should like to quote the author Ian Sinclair from this excellent book "SIMPLE 'C' A Beginners guide' published by David Fulton Publishers

C is a language which combines the power of machine-code with the structure of a high level language.

C allows you the control over what the computer does that you normally associate with machine-code and will generate compiled code that is almost as compact as machine-code from an assembler At the same time, though, C provides all the structures of a good high level language, like facilities for creating loops, many different variable types, structured variables like arrays, and so on

Next issue I shall begin with getting down to the nitty gritty of actually writing programs in C The package I shall be using for this will be the highly recommended "Spinnakker's POWER C" for the Commodore 64/128

SOME RECOMMENDED READING,

The C Programming Language Kernmingham & Ritchie (Prentice-Hall)

The C Programming Tutor. Wortman & Sidebottman (Prentice-Hall)

The C Programmers Handbook Thom Hogan (Prentice-Hall)

Simple 'C' A Beginner s Guide Ian Sinclair (David Fulton

BINDERS

£6.80 Inc P&P

FOR YOUR VALUABLE COLLECTION OF

COMMODORE DISK USER

MAGAZINES

B I N D E R S

★ TOP QUALITY

★ SMART

★ EASY TO USE

ASP
READER SERVICES
ARGUS HOUSE
BOUNDARY WAY
HEMEL HEMPSTEAD
HERTS HP2 7ST
Telephone your order
(0442) 66551

Please supply .. **COMMODORE DISK USER BINDERS** @ £6 80 each inc p&p

Total £ . (Please make cheques/postal orders payable to A S P)

NAME

ADDRESS

Or debit my ACCESS/VISA ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Expiry .

Please allow 28 days for delivery

Power Cartridge Competition Results

Our Editor wades through the mountain of entries in the recent competition to find the 25 winners

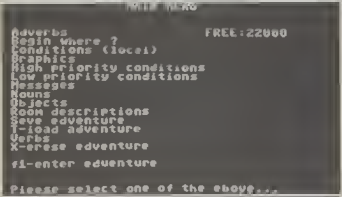
The recent 'Power Cartridge' competition proved to be very successful with all our readers The entries came in by the sack load As always, there has to be losers as well as winners If your name is not one of the lucky ones we are very sorry

List Of Prize Winners

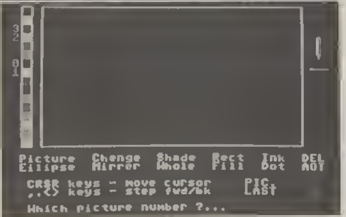
A G.Taylor Essex	P K Smith Notts
M Mitchell Derbyshire	H.Samson Middlesex
N. Gregory Cumbria	S Knipe Flamborough
G McNaughton Perth	A Speight Hull
W Burton Barnsley	G.Cooper Totton
F.Burgess Spalding	M Dexter Grays
C Woodhouse Hull	R.Underwood Subiton
M Mitchell London	W.Brown Tidworth
D Washer Dubai	I Bradey Sheffield
B Young Perth	J AWayman Letchworth
J W Borland London	M McLean Glasgow
M Smith Southampton	C.Henocq France
A Mitchell Barrow-on-Humber	

Congratulations to all the above. Your prize will be despatched as soon as possible Once again I would like to thank you all for taking the time and effort to send in your entry forms

G.A.C. + under review



```
MAIN MENU

Adverbs                              FREE:22000
Begin where ?
Conditions (local)
Graphics
High priority conditions
Low priority conditions
Messages
Nouns
Objects
Room descriptions
Save adventure
T-load adventure
Verbs
X-erase adventure

f1-enter adventure

Please select one of the above...
```



```
Picture  Change  Shade  Rect  Ink  DEL
Ellipse  Mirror  Whole  Fill  Dot  NOT

CRSR keys - move cursor      PIC
,.<> keys - step fwd/bk      LAST

Which picture number ?...
```

Tony Rome investigates the updated G A C system and gives us this report

To the would be adventurer, both beginner and expert, the very thought of setting off on a dangerous mission, risking all for the world's largest diamond, the conquest of space or the hand of a princess, is both compelling and addictive!

Personally, having used this programme, I found it easy to follow The instructions are explicit and uncomplicated and the final results can be quite effective Incentive Software state as follows, The picture editor is 'packed with various features allowing you to easily create stunning graphics to illustrate locations.' This I found to be true, depending upon your ability as an artist G.A.C. has been on the market for a few years now but technically still holds its own as an adventure writing package

The main features include automatic word formatting, an intelligent command interpreter, abbreviated input acceptance, synonym recognition, space for 765 nouns, verbs and adverbs, multiple command lines, recognition up to full length of word, 'it' detection, extensive text compression etc and a 'save and load position from a game' facility etc.

The 'PLUS' incorporated in the new G.A.C Plus is an addition to the old programme giving it virtually unlimited scope and enabling the writer to compose large scale multi-linked adventures. This in effect means that you are able to freely move from an original adventure into other adventures and back without using noticeable memory which is drawn instead from saved data on a disc. This allows you a more comprehensive vocabulary which is essential to the well constructed storyline

Everything in G A C + is clearly mapped out in the instruction manual and can be easily comprehended.

Returning to the 'Graphic' side of G.A.C +. Graphics like these are made simpler using the various 'tools' of the picture editor i e (Shading, Rectangle, Ellipse, Dot and many other features all constructed on a hi-resolution screen.

If you bought the December issue of C D.U. you will know that 'KRON' was achieved using the Graphic Adventure Creator. This is an illustration of G A.C s ability

For those of you that already have G.A.C. you may be pleased to hear of a software package written by Don Macleod. This programme will give you a comprehensive printout of Room Descriptions, messages, Nouns, Verbs, Adverbs, Objects and Conditions. It also has the facility to delete unwanted letters and words that you have erased but remain in the programme! This is an invaluable addition saving your hours of searching through screen listings trying to find out why the 'Gorilla ate the latter instead of a banana', or why you can't open the vault!

This programme which is called the G A.C Database Printer is available from BIG SKY SOFTWARE – more details later So. if you think you can write a marketable adventure then all in all G.A.C + is well worth its fairly modest price

By the way, if you enjoyed 'KRON' then look out for THE CRAMORE DIAMOND CAPER, coming soon in CDU and using the G.A.C Database Printer – Price £5 available from:-

BIG SKY SOFTWARE
35 Old Evanton Road
DINGWALL
Rosshire IV15 9RB
SCOTLAND

G.A C.+ Available from:-
Incentive Software Ltd
Exclusive Distributor
Mandy Rodrigues
67 Lloyd St
LLANDUDNO
Gwynedd
LL20 2YP '
G.A.C.+ £29 95 or send £10 together with your old G.A.C PROGRAMME (Disk or Cassette) for the updated PLUS version.

Continued from page 19

Animating the sprite

To animate the sprite you must store the various frames of animation in consecutive memory locations. Then press A A prompt in the drawing area will ask for the starting pointer, enter the pointer for the first frame of animation in hex, then a prompt will ask for the final pointer, enter the pointer for the last frame again in hex The animation will then be displayed in the top right hand sprites and also in the current library sprite. The current pointer that is being animated is shown in the drawing area during animation. The animator will exit to the main program when the sequence is finished or if any key is pressed during animation.

Input/output

Pressing '@' will display the Disk Menu. The Disk Menu allows all saving, loading, directorying etc It consists of the following options.

1. Load
2. Save
3. Verify
4. Directory.
5. Disk Status
6. DOS Command.
7. Exit

1: This allows the loading of data A prompt will ask for the device (1/8) where 1 is tape and 8 is disk Then it will ask for a filename (no filename is necessary on tape). Another prompt will ask 'Default Address (Y/N) ?' this is to find out if you want the program loaded at the area it was saved or whether you want it loaded into a new location. If 'N' is entered you will be asked for the new address Otherwise it will be loaded into the area it was saved from. Finally you will be asked if everything is correct where 'Y' will continue and 'N will abort.

2 This allows the saving of data It will then ask for the device and filename as above You will then be prompted for the start and end addresses of the save Finally you will be asked to confirm the information as above. Entering @0:Filename will overwrite the existing named file.

3: This allows you to verify data that has been saved. This will ask for the device and filename and then it will verify

4: This displays the disk directory

5: This displays the disk status. Ie if there is a disk error.

6: This allows you to enter DOS commands some of which are listed below The full range of DOS commands and errors are listed in the disk drive manual.

COMMAND		TYPE
Directory		$
Disk Status		@
New (Format)		N Diskname,ID
Rename		R Newname=Oldname
Scratch		S.Filename
Initialise		I
Validate		V

7: This exits to the main program.

Helpscreens

Helpscreens are available by pressing H Press any key to view the second screen and then to exit

Key summary

The keys come in order from the top left of the keyboard

KEY	USE
1	Library Sprite 1
2	Library Sprite 2
3	Library Sprite 3
4	Library Sprite 4
5	Library Sprite 5
[illegible]	Change Sprint Bank
[illegible]	Increment Pointer
[illegible]	Decrement Pointer
CLR/HOME	
SHIFT	Clear Current Sprite and Drawing Area
INST/DEL	Change Text Colour
W	[illegible]pe [illegible] Sprite
[illegible]	Exclus[illegible] (mono Only
[illegible]	Scroll Right
[illegible]	Trade current sprite wi current sprint sprite
[illegible]	Y mirror
[illegible]	Scroll UP
[illegible]	Position current libra sprite
@	Disk menu
[illegible]	Animate
[illegible]	Store current sprite at current sprint positio
[illegible]	Scroll DOWN
[illegible]	Grab current sprint sprite and place it in the drawing area and also in the current library sprite
[illegible]	Scroll LEFT
[illegible]	Increment sprint colo
[illegible]	Multicolour/monochrome sprint toggle
RETURN	Plot current colour at cursor position
X	X Mirror
C	X Expansion of curre library sprite and spri in top right corner of screen
V	Y Expansion of current library sprite and sprit in top right corner of screen
[illegible]	Increment border colour
N	Selector for current col our in multicolour mod
M	Multicolour/monochrome sprite toggle
[illegible]	Increment multicolour
[illegible]	Increment multicolour
CURSOR KEYS-UP	Move cursor UP
DOWN	Move cursor DOWN
LEFT	Move cursor LEFT
RIGHT	Move cursor RIGHT
SPACE BAR	Delete at current curs position
JOYSTICK IN PORT 2	Same as cursor keys
FIRE BUTTON	Same as return or space depending on drawing mode
F1	Select draw
F3	Select delete
F5	Increment backgroun colour
[illegible]	Increment sprite colo

Disk Dungeons

Gordon Hamlett looks at what's new on the scene for adventurers.

This is still a very quiet period with only one new game to report on which, unfortunately, I didn't enjoy very much but you can't have everything It will be interesting to see whether this lull is only temporary or whether all current role playing game (RPG) work is being done largely for the sixteen bit market with eight bit conversions only as an afterthought.

The only new games that are being heavily advertised at the moment are the latest Dungeons and Dragons games from SSI. These games are based on the hugely popular Dragonlance series of books and will feature a war game and arcade games as well as RPGs. The most successful RPGs are still fantasy based and I started musing on the reasons for this. Certainly, there is scope for a good game that doesn't involve trolls and treasure but no-one seems to be writing it at the moment! There has been a mini spate of science fiction stories recently but these just don t seem to work as well. Combat with lasers, ray guns and photon torpedoes is no substitution for swords and slings. A message informing you that your ship's shields are down to 13% efficiency seems so much more impersonal than knowing that your favourite wizard only has three hit points left! Another factor is that sci-fi combat tends to offer less scope for tactics than the traditional hand to hand stuff There's not a lot you can do with 'Zap– You're dead!' With the whole field of magic missing as well, two of the main pillars of the fantasy RPG have been removed at one fell swoop.

What do you think? Am I talking through the end of my body not normally associated with speech as usual or do you still prefer fantasy games? If not, what sort of stories would you like to see instead? Pirates, Westerns, Yuppies or what Write in and tell me what you really want to play You never know, there might be a company out there somewhere who is looking for a decent original plot.

Enough of the waffle for this month, and on to the review. Sentinel Worlds 1 – Future Magic from Electronic Arts

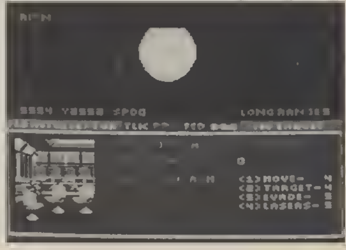
SENTINEL WORLDS 1

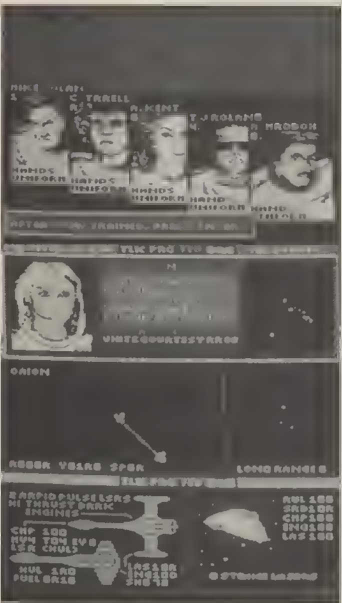
You have been commissioned by the Federation to investigate a series of raids on Cargoliners which were ferrying supplies from the Caldorre System to outlying frontiers. The intelligence available to you is minimal if not to say non-existent as you and your crew of five set off on a mission from which you are not expected to return

There is a crew already picked for you if you want to start play straight away or you can start from scratch and train the crew just the way you want them. The five members are pilot, navigator, communications officer, engineer and medic. Each crew member has five basic characteristics – strength, stamina, dexterity, comprehension and charisma and you assign between 10 and 20 points to these traits as you see fit from a central well of 70 points

In addition, each crew member has a series of additional skills; in weapons (contact, edged, projectile and blasters), tactics, reconnaissance, gunnery, repair of the all terrain vehicle, mining, athletic prowess, observation and bribery It is up to you to get the balance of skills correct for your party as no one person can do everything. As you become more experienced, promotion comes and so you can enhance existing skills or learn new ones

Once in space, you are quickly attacked and your convoy broken up.





From now on, you're on your own. You can either stay in space attacking the pirate ships and discovering what you can from boarding them or explore the different planets and space stations with the intention of gleaning extra intelligence there. The main problem with this game lies in the space travel which is very much a hit or miss affair, certainly in the early stages. The scanner display is almost impossible to read – tiny little dots which may or may not be different colours, (it is difficult to tell with the flickering) and finding your way around is fairly random. Why you couldn't be provided with even the co-ordinates of all friendly bases let alone a sophisticated system such as locking onto a beacon, is beyond me. And it really does spoil the game.

Similarly, moving your spaceship around is a fairly crude affair and enemy craft whizz past you as you try to manoeuvre. For anyone who likes to take their time and ponder each move, this 'real-time' action will be another aspect of the game that won't appeal to them.

As a result of all this, I found the game very difficult to get into and soon lost all enthusiasm for it. This is a pity as a cursory look at the packaging shows that there is a lot of substance here for those who persevere with tunnels, planets and enemy spaceships to explore. I can't help feeling that it would have been a lot better to start the adventure off on land and give the player a chance to explore and get a feel for the game before casting him or her off into deep space.

A lot of these criticisms are tied in with what I feel about science fiction role playing games in general (see above). At the moment, they just don't grab me when I desperately want to be grabbed. There is too much space flight which tends to be extremely boring and time consuming. Can't we have a futuristic game that is just set on one planet? I seem to remember that Wasteland managed to do that and it is surely no coincidence that that is still the best sci-fi game to date.

Finally this month, a damsel in distress, namely Mrs J. Hatchett from Brighton who is having trouble at the start of Ultima V in so much as she keeps getting beaten up by the Shadowlords when she visits a town and doesn't have sufficient funds to buy herself any decent equipment.

The Shadowlords are designed to be a major menace and the obvious suggestion is to explore the places where they aren't first! So, concentrate your early explorations around Lord British' castle, the town of Britain and the Brittanys, villages, lighthouses and Castles. That should keep you quiet for a bit.

As far as money and supplies go, walk backwards across bridges until you get attacked by trolls. Defeating them normally provides a good haul of treasure which you can sell or use as you see fit.

COMMODORE DISK·USER·

Linsags: 53p psr word. (+ VAT)
Semi display: £11.50 plus VAT psr single column centimstre minimum 2cm. Ring for information on series bookings/discounts.
All advertisemsnts in this section must bs prepaid.
Advertisemsnts srs accepted subject to the tsrms and conditions printsd on ths advertisemsnt rats card (svailabls on request).
Meks chsquee peyeble to ASP Ltd.

Send your requirements to:
CLASSIFIED DEPARTMENT
ASP LTD. ARGUS HOUSE,
BOUNDARY WAY, HEMEL HEMPSTEAD HP2 7ST.

0442 66551

DISKS

WESTONING LTD

3½ DISKS		5¼ DISKS		AMIGA	
DS/DD . .	46p	DS/DD .	.23p	BATMAN PACK	£355
DS/HD . . .	. . .£1.15	DS/HD . . .	. . . .57p	FLIGHT FANTASY PK	.£355
				CLASS OF 90 s	£510
				512K RAM	.£56
				WITH CLOCK . . .	.£62

PRICE EACH INC VAT AND LABELS — PRICE EACH INC VAT SLEEVES AND LABELS

LOCKABLE BOXES		LOCKABLE BOXES	
CAP 40 - £4	CAP 80 - £5.75	CAP 50 - £5.25	CAP 100 - £6.50

NEC EXTERNAL
3½ DRIVE .£60
1084D MON .£239

PRICES INCLUDE VAT NO QUIBBLE GUARANTEE
POST ADD £3 NEXT DAY ADD £7

12 SANDERSON ROAD, WESTONING,
BEOFORD MK45 5JY
Tele. (0636) 775060 or (0525) 716668 Fex (0525) 715789

REPAIRS

M.C.E. SERVICES

33 Albert Street, Mansfield, Notts. NO19 1EA Tel 0623 653512

FIXED PRICE COMMODORE REPAIRS

C64	£24 99
C16	£19.95
PLUS4	£24 95
AMIGA A500	£39.95
1541	£29 95
1571	£31 99
C2N etc	£12.95

All prices inc. VAT, Labour, Parts etc.
NO MORE TO PAY
Please ring for details.

COMMODORE REPAIRS, SPARES & LEAOS

C64 +4 16 - £24 00 C128 - £32.00
AMIGA 500 -£34.00
Three month warranty For price list send 1st class stamp (State model)

ACE REPAIRS, Outweys FM, Pelynt, Looe, Cornwell, PL13 2NW. Tel. (0503) 20282

COMMODORE SPARES & REPAIRS

REPAIRS

C64, C+4	.£25 Inc.
C128, 1541 .	£35 Inc.

SPARES

906114	£9 30	901227	£11 95
4510	£9 95	8501	£11 85
4526	£10 95	C64 Power Pack	£22 00
4561	£12 95	C64 Cassette Unit	£27 00
6569	£15 95	C128 Power Pack	£69 95

Inc Vat & PP

OASIS COMPUTER SERVICES,
Dept 4D, 14 Ridgeway Road, Salisbury, Wiltshire, SP1 3BU.
Tel (0722) 335061

TO ADVERTISE IN CDU CALL TONY FLANAGAN ON 0442 66551

SOFTWARE

PUBLIC DOMAIN SOFTWARE for the C64/128

We have 150 disks full of all types of Programs
All disks are £2 95 each
NOW FOR THE 4
from £3 75 per disk
Send SAE or phone for free catalogue (state model)
Kingsway Computer Services
72 Glencoe Road Sheffield
S2 2SR Tel (0742) 750623

COMPLETE THE COUPON BELOW OR CALL 0442 66551 TO ADVERTISE

SERVICES

COMMODORE SUPPLIES

C64 Power Unit	£21 99
C128 Power Unit	£39 99
C2N/C3N Datasette	£24 99
100 x 3½" Discs	£29 99
C64 Magnum Lightgun	£34.99
C64/128 Serial Cable	£5.99

Prices include VAT & P&P Chq/PO to
OMNIDALE SUPPLIES (CO),
23 Curzon Street, Derby, DE1 2ES
Tel 0332 291219

COMMODORE DISK·USER·

Lineage **53p** per word (+ VAT)
Semi Display **£11.50** (+ VAT) per single column centimetre Min size 2cm.
Series discounts available on request All classified advertisements must be pre-paid
There are no reimbursements for cancellations
SENO TO CDU ADVERTISEMENT DEPARTMENT, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HP2 7ST

I enclose my Cheque/Postal Order lor £ . . . . . lor . . . . insertions, made payeble to Argus Specialist Publications (Delete as necessary)

PLEASE OEBIT MY ACCESS/BARCLAYCARD NO

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ EXP. DATE . . . .

£ . . . . FOR . . . . .INSERTIONS

Neme . . . . Postcode . . . .

Deytime Tel No. . . . . Signature . . . . Oete . . . .

☐ FOR SALE ☐ SOFTWARE ☐ SPECIAL OFFERS ☐ REPAIRS ☐ NAROWARE ☐ OISKS

...it's dynamite!

POWER CARTRIDGE

FOR YOUR COMMODORE 64/128

SO MUCH FOR SO LITTLE

"...unbelievable value for money" ZZAPP! Dec 89

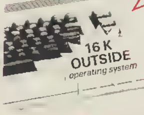
* POWER TOOLKIT
* POWER MONITOR
* TAPE & DISK TURBO
* PRINTERTOOL
* POWER RESET
* TOTAL BACKUP

..."Money well spent" YC/CDU Jan 90

42 Pg Manual - "Damned Good Handbook" CCI Jan 90

AVAILABLE FROM ALL GOOD COMPUTER RETAILERS

TRIED AND TESTED - OVER 100,000 SOLD IN EUROPE



..."highly recommended for C64 users" CCI Jan 90

YOU WILL WONDER HOW YOU EVER MANAGED WITHOUT IT

ONLY £16.99 incl. VAT

[illegible]

A powerful BASIC Toolkit (Additional helpful commands) that considerably simplifies programming and debugging

AUTO	HARDCAT	RENUMBER
AUDIO	HARDCOPY	REPEAT
COLOR	HEX$	SAFE
DEEK	INFO	TRACE
DELETE	KEY	UNNEW
DOKE	PAUSE	QUIT
DUMP	PLIST	MONITOR
FIND	LOAD	BLOAD

RENUMBER	Also modifies all the GOTO's GOSUB's etc. Allows part of a program to be renumbered or displaced
PSET	Set up of printer type
HARDCAT	Prints out Directory.

The toolkit commands can be used in your programs.

[illegible]

Using POWER CARTRIDGE you can load up to 6 times faster from disk.
The Disk commands can be used in your own programs.

DLOAD	DVERIFY	DIR
DSAVE	MERGE	DEVICE
DISK		

MERGE	Two BASIC programs can be merged into one
DISK	With DISK you can send commands directly to your disk

[illegible]

Using POWER CARTRIDGE you can work up to 10 times faster with your data recorder. The Tape commands can be used in your own programs.

LOAD	SAVE	VERIFY
MERGE	AUDIO	

[illegible]

A powerful machine language monitor that is readily available and leaves all of your Commodore memory available for programming.
Also works in BASIC ROM, KERNAL and I/O areas.

A ASSEMBLE	I INTERPRET	S SAVE
C COMPARE	J JUMP	T TRANSFER
D DIS-ASSEMBLE	L LOAD	V VERIFY
F FILL	M MEMORY	W WALK
G GO	P PRINT	X EXIT
H HUNT	R REGISTER	$ DIRECTORY
		DOS Commands

[illegible]

The POWER CARTRIDGE contains a very effective Printer Interface that self detects if a printer is connected to the Serial Bus or User Port.
It will print all Commodore characters on Epson and compatible printers.
The printer interface has a variety of set-up possibilities. It can produce HARDCOPY of screens not only on Serial printers (MPS801, 802, 803 etc) but also on Centronic printers (EPSON, STAR, CITIZEN, PANASONIC, etc).
The HARDCOPY function automatically distinguishes between HIRES and LORES. Multi-colour graphics are converted into shades of grey. The PSET functions allow you to decide on Large/Small and Normal/Inverse printing.
The printer PSET functions are:

PSET 0	Self detection Serial/Centronics
PSET 1	EPSON mode only
PSET 2	SMITH-CORONA mode only
PSET 3	Turns the printing 90 degrees!!
PSET 4	HARDCOPY setting for MPS802/1526
PSET B	Bit-image mode
PSET C	Setting Lower/Upper case and sending Control Codes
PSET T	All characters are printed in an unmodified state
PSET U	Runs a Serial printer and leaves the User port available
PSET Sx	Sets the Secondary address for HARDCOPY with Serial Bus
PSET L1	Adds a line-feed, CHR$ (10), after every line
PSET L0	Switches PSET L1 off

[illegible]

On the back of the POWER CARTRIDGE there is a Reset Button. Pressing this button makes a SPECIAL MENU appear on the screen.
This function will work with many programmes.

CONTINUE	Allows you to return to your program
BASIC	Return to BASIC
RESET	Normal RESET
TOTAL BACKUP DISK	Saves the contents of the memory onto a Disk. The program can be reloaded later with BLOAD followed by CONTINUE
RESET ALL	RESET of any program
TOTAL BACKUP TAPE	As BACKUP DISK but to TAPE
HARDCOPY	At any moment, prints out a Hardcopy of the screen. Using CONTINUE afterwards you can return to the program
MONITOR	Takes you into the Machine language Monitor

BDL

Bitcon Devices Ltd

88 BEWICK ROAD
GATESHEAD
TYNE AND WEAR
NE8 1RS
ENGLAND

Tel: 091 490 1975 and 490 1919 Fax 091 490 1918
To order: Access/Visa welcome - Cheques or P/O payable to BDL
Price: £16.99 incl. VAT.
UK orders add £1.20 post/pack total - £18.19 incl. VAT.
Europe orders add £2.50 Overseas add £3.50
Scandinavian Mail Order and Trade enquiries to: Bhiab Elektronik, Box 216, Norrtelje 76123, SWEDEN. Tel: + 46 176 18425 Fax. 176 18401
TRADE AND EXPORT ENQUIRIES WELCOME